MIST
MULTIPLE INSTANCE SPATIAL TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a deep network that can be trained to tackle image reconstruction and classification problems that involve detection of multiple object instances, *without* any supervision regarding their whereabouts. The network learns to extract the most significant K patches, and feeds these patches to a task-specific network — e.g., auto-encoder or classifier — to solve a domain specific problem. The challenge in training such a network is the non-differentiable top-K selection process. To address this issue, we lift the training optimization problem by treating the result of top-K selection as a slack variable, resulting in a simple, yet effective, multi-stage training. Our method is able to learn to detect recurring structures in the training dataset by learning to reconstruct images. It can also learn to localize structures when only knowledge on the occurrence of the object is provided, and in doing so it outperforms the state-of-the-art.

1 INTRODUCTION

Finding and processing multiple instances of characteristic entities in a scene is core to many computer vision applications, including object detection (Ren et al., 2015; He et al., 2017; Redmon & Farhadi, 2017), pedestrian detection (Dollár et al., 2012; Sewart & Andriluka, 2016; Zhang et al., 2018a), and keypoint localization (Lowe, 2004; Bay et al., 2008). In traditional vision pipelines, a common approach to localizing entities is to select the top-K responses in a heatmap and use their locations (Lowe, 2004; Bay et al., 2008; Felzenszwalb et al., 2010). However, this type of approach does not provide a gradient with respect to the heatmap, and cannot be directly integrated into neural network-based systems. To overcome this challenge, previous work proposed to use grids (Redmon et al., 2016; He et al., 2017; Detone et al., 2018) to simplify the formulation by isolating each instance (Yi et al., 2016), or to optimize over multiple branches (Ono et al., 2018). While effective, these approaches require additional supervision to localize instances, and do not generalize well outside their intended application domain. Other formulations, such as sequential attention (Ba et al., 2015; Gregor et al., 2015; Eslami et al., 2015) and channel-wise approaches (Zhang et al., 2018c) are problematic to apply when the number of instances of the same object is large.

Here, we introduce a novel way to approach this problem, which we term *Multiple Instance Spatial Transformer*, or *MIST* for brevity. As illustrated in Figure 1 for the image synthesis task, given an image, we first compute a heatmap via a deep network whose local maxima correspond to locations of interest. From this heatmap, we gather the parameters of the top-K local maxima, and then extract the corresponding collection of image patches via an image sampling process. We process each patch independently with a task-specific network, and aggregate the network's output across patches.

Training a pipeline that includes a non-differentiable selection/gather operation is non-trivial. To solve this problem we propose to lift the problem to a higher dimensional one by treating the parameters defining the interest points as slack variables, and introduce a hard constraint that they must correspond to the output that the heatmap network gives. This constraint is realized by introducing an auxiliary function that creates a heatmap given a set of interest point parameters. We then solve for the relaxed version of this problem, where the hard constraint is turned into a soft one, and the slack variables are also optimized within the training process. Critically, our training strategy allows the network to incorporate both non-maximum suppression and top-K selection. We evaluate the performance of our approach for ① the problem of recovering the basis functions that created a given texture,

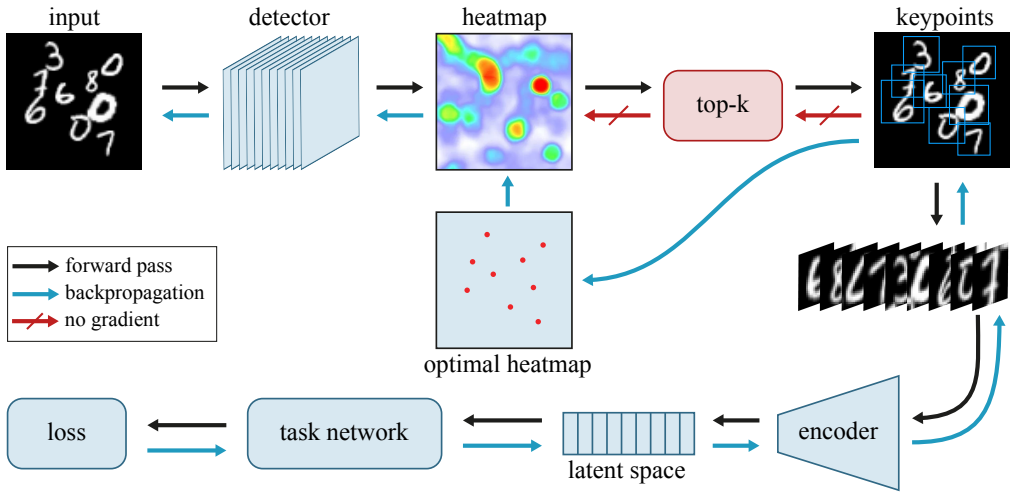

Figure 1: **The MIST architecture** – A network $\mathcal{H}_\eta$ estimates locations and scales of patches encoded in a heatmap $\mathbf{h}$. Patches are then extracted via a sampler $\mathcal{S}$, and then fed to a task-specific network $\mathcal{T}_\tau$. In this example, the specific task is to re-synthesize the image as a super-position of (unknown and locally supported) basis functions.

② classification of handwritten digits in cluttered scenes, and ③ recognition of house numbers in real-world environments. In summary, in this paper we:

- introduce the MIST framework for weakly-supervised multi-instance visual learning;
- propose an end-to-end training method that allows the use of top-K selection;
- show that our framework can reconstruct images as parts, as well as detect/classify instances without any location supervision.

2 Related work

Attention models and the use of localized information have been actively investigated in the literature. Some examples include discriminative tasks such as fine-grained classification (Sun et al., 2018) and pedestrian detection (Zhang et al., 2018a), and generative ones such as image synthesis from natural language (Johnson et al., 2018). We now discuss a selection of representative works, and classify them according to how they deal with multiple instances.

Grid-based methods. Since the introduction of Region Proposal Networks (RPN) (Ren et al., 2015), grid-based strategies have been used for dense image captioning (Johnson et al., 2016), instance segmentation (He et al., 2017), keypoint detection (Georgakis et al., 2018), multi-instance object detection (Redmon & Farhadi, 2017). Recent improvements to RPNs attempt to learn the concept of a generic object covering multiple classes (Singh et al., 2018), and to model multi-scale information (Chao et al., 2018). The multiple transformation corresponding to separate instances can also be densely regressed via Instance Spatial Transformers (Wang et al., 2018), which removes the need to identify discrete instance early in the network. However, all these methods are fully supervised, requiring both class *labels* and object *locations* for training.

Heatmap-based methods. Heatmap-based methods have recently gained interest to detect features (Yi et al., 2016; Ono et al., 2018; Detone et al., 2018), find landmarks (Zhang et al., 2018c; Merget et al., 2018), and regress human body keypoint (Tekin et al., 2017; Newell et al., 2016). While it is possible to output a separate heatmap per class (Zhang et al., 2018c; Tekin et al., 2017), most heatmap-based approaches do not distinguish between instances. Yi et al. (2016) re-formulate the problem based on each instance, but in doing so it introduces a non-ideal difference between training and testing regimes. Grids can also be used in combination with heatmaps (Detone et al., 2018), but this results in an unrealistic underlying assumption of uniformly distributed detections in the image. Overall, heatmap-based methods excel when the "final" task of the network is generate a heatmap (Merget et al., 2018), but are problematic to use as an intermediate layer in the presence of multiple instances.

Sequential inference methods. Another way to approach multi-instance problems is to attend to one instance at a time in a sequential way. Training neural network-based models with sequential attention is challenging, but approaches employing policy gradient (Ba et al., 2015) and differentiable attention mechanisms (Gregor et al., 2015; Eslami et al., 2015) have achieved some success for images comprising *small* numbers of instances. However, RNNs often struggle to generalize to sequences longer than the ones encountered during training, and while recent results on inductive reasoning are promising (Gupta et al., 2018), their performance does not scale well when the number of instances is large.

Knowledge transfer. To overcome the acquisition cost of labelled training data, one can transfer knowledge from labeled to unlabeled dataset. For example, Inoue et al. (2018) train on a single instance dataset, and then attempt to generalize to multi-instance domains, while Uijlings et al. (2018) attempt to also transfer a multi-class proposal generator to the new domain. While knowledge transfer can be effective, it is highly desirable to devise unsupervised methods such as ours that do not depend on an additional dataset.

Weakly supervised methods. To further reduce the labeling effort, weakly supervised methods have also been proposed. Wan et al. (2018) learn how to detect multiple instances of a single object via region proposals and ROI pooling, while Tang et al. (2018) propose to use a hierarchical setup to refine their estimates. Gao et al. (2018) provides an additional supervision by specifying the number of instances in *each* class, while Zhang et al. (2018b) localize objects by looking at the network activation maps (Zhou et al., 2016; Selvaraju et al., 2017). Shen et al. (2018) introduce an adversarial setup, where detection boxes are supervised by distribution assumptions and classification objectives. However, all these methods still rely on region proposals from an existing method, or define them via a hand-tuned process.

3 MIST FRAMEWORK

A prototypical MIST architecture is composed of two trainable components: ① the first module receives an image as input and extracts a collection of patches, at image locations and scales that are computed by a trainable heatmap network $\mathcal{H}_\eta$ with weights η; see Section 3.1. ② the second module processes each extracted patch with a task-specific network $\mathcal{T}_\tau$ whose weights τ are shared across patches, and further manipulates these signals to express a task-specific loss $\mathcal{L}_{task}$; see Section 3.2. The two modules are connected through non-maximum suppression on the scale-space heatmap output of $\mathcal{H}_\eta$, followed by a top-K selection process to extract the parameters defining the patches, which we denote as $\mathcal{E}_K$. We then sample patches at these locations through bilinear sampling $\mathcal{S}$ and feed them the second module.

The defining characteristic of the MIST architecture is that they are *quasi-unsupervised*: the only strictly required supervision is the number K of patches to extract. The training of the MIST architecture is summarized by the optimization:

$$\underset{\tau,\eta}{\operatorname{argmin}} \ \mathcal{L}_{\text{task}}(\mathcal{T}_\tau(\mathcal{S}(\mathcal{E}_K(\mathcal{H}_\eta(\mathcal{I}))))) \tag{1}$$

where τ, η are the network trainable parameters. In this section, we describe the forward pass through the MIST architecture. Because the patch extractor $\mathcal{E}_K$ is non-differentiable, optimizing this objective presents additional challenges, which we address in Section 4.

3.1 PATCH EXTRACTION

We extract a set of K (square) patches that correspond to "important" locations in the image – where importance is a direct consequence of $\mathcal{L}_{\text{task}}$. The localization of such patches can be computed by regressing a 2D heatmap whose top-K peaks correspond to the patch centers. However, as we do not assume these patches to be equal in size, we regress to a collection of heatmaps at different scales. To limit the number of necessary scales, we use a discrete scale space with S scales, and resolve intermediate scales via weighted interpolation.

Multiscale heatmap network – $\mathcal{H}_\eta$. Our multiscale heatmap network is inspired by LF-Net (Ono et al., 2018). We employ a fully convolutional network with (shared) weights η at multiple scales, indexed by $s = 1 \dots S$, on the input image $\mathcal{I}$. The weights η across scales are shared so that the

network cannot implicitly favor a particular scale. To do so, we first downsample the image to each scale producing $\mathcal{I}_s$, execute the network $\mathcal{H}_\eta$ on it, and finally upsample to the original resolution. This process generates a multiscale heatmap tensor $\mathbf{h} = \{\mathbf{h}_s\}$ of size $H \times W \times S$ where $\mathbf{h}_s = \mathcal{H}_\eta(\mathcal{I}_s)$, and H is the height of the image and W is the width. For the convolutional network we use 4 ResNet blocks (He et al., 2015), where each block is composed of two 3×3 convolutions with 32 channels and relu activations without any downsampling. We then perform a *local spatial softmax* operator (Ono et al., 2018) with spatial extent of 15×15 to sharpen the responses. Then we further relate the scores across different scales by performing a "softmax pooling" operation over scale. Specifically, if we denote the heatmap tensor after local spatial softmax as $\tilde{\mathbf{h}} = \{\tilde{\mathbf{h}}_s\}$, since after the local spatial softmax $\mathcal{H}_\eta(\mathcal{I}_s)$ is already an "exponentiated" signal, we do a weighted normalization without an exponential, *i.e.* $\mathbf{h}' = \sum_s \tilde{\mathbf{h}}_s(\tilde{\mathbf{h}}_s / \sum_{s'}(\tilde{\mathbf{h}}_{s'} + \epsilon))$, where $\epsilon = 10^{-6}$ is added to prevent division by zero.

Top-K patch selection – $\mathcal{E}_K$. To extract the top K elements, we perform an addition cleanup through an actual non-maximum suppression. We then find the spatial locations of the top K elements of this heatmap $\bar{\mathbf{h}}_s$, denoting the spatial location of the k^{th} element as (x_k, y_k), which now reflect local maxima. For each location, we compute the corresponding scale by weighted first order moments (Suwajanakorn et al., 2018) where the weights are the responses in the corresponding heatmaps, *i.e.* $s_k = \sum_s \mathbf{h}'_s(x_k, y_k)s$.

Our extraction process uses a single heatmap for all instances that we extract. In contrast, existing heatmap-based methods (Eslami et al., 2015; Zhang et al., 2018c) typically rely on heatmaps dedicated to *each* instance, which is problematic when an image contains two instances of the same class. Conversely, we restrict the role of the heatmap network $\mathcal{H}_\eta$ to find the "important" areas in a given image, without having to distinguishing between classes, hence simplifying learning.

Patch resampling – $\mathcal{S}$. As a patch is uniquely parameterized its location and scale $\mathbf{x}_k = (x_k, y_k, s_k)$, we can then proceed to resample its corresponding tensor via bilinear interpolation (Jaderberg et al., 2015; Jiang et al., 2019) as $\{\mathbf{P}_k\} = \mathcal{S}(\mathcal{I}, \{\mathbf{x}_k\})$.

3.2 TASK-SPECIFIC NETWORKS

We now introduce two applications of the MIST framework. We use the *same* heatmap network and patch extractor for both applications, but the task-specific network and loss differ. We provide further details regarding the task-specific network architectures in Section B of the supplementary material.

Image reconstruction / auto-encoding. As illustrated in Fig. 1, for image reconstruction we append our patch extraction network with a *shared* auto-encoder for each extracted patch. We can then train this network to *reconstruct* the original image by inverting the patch extraction process and minimizing the mean squared error between the input and the reconstructed image. Overall, the network is designed to *jointly* model and localize repeating structures in the input signal. Specifically, we introduce the generalized inverse sampling operation $\mathcal{S}^{-1}(\mathbf{P}_i, \mathbf{x}_i)$, which starts with an image of all zeros, and places the patch $\mathbf{P}_i$ at $\mathbf{x}_i$. We then sum all the images together to obtain the reconstructed image, optimizing the task loss

$$\mathcal{L}_{\text{task}} = \left\| \mathcal{I} - \sum_i \mathcal{S}^{-1}\left(\mathbf{P}_i, \mathbf{x}_i\right) \right\|_2^2. \tag{2}$$

Multiple instance classification. By appending a classification network to the patch extraction module, we can also perform multiple instance learning. For each extracted patch $\mathbf{P}_k$ we apply a shared classifier network to output $\hat{\mathbf{y}}_k \in \mathbb{R}^C$, where C is the number of classes. In turn, these are then converted into probability estimates by the transformation $\hat{\mathbf{p}}_k = \text{softmax}(\hat{\mathbf{y}}_k)$. With $\mathbf{y}_l$ being the one-hot ground-truth labels of instance l, we define the multi-instance classification loss as

$$\mathcal{L}_{\text{task}} = \left\| \frac{1}{L} \sum_{l=1}^{L} \mathbf{y}_l - \frac{1}{K} \sum_{k=1}^{K} \hat{\mathbf{p}}_k \right\|_2^2, \tag{3}$$

where L is the number of instances in the image. We empirically found this loss to be more stable compared to the cross-entropy loss, similar to Mao et al. (2017). Note here that we *do not* provide supervision about the localization of instances, yet the detector network will automatically learn how to localize the content with minimal supervision (i.e. the number of instances).

Algorithm 1 Multi-stage optimization for MISTs

Require: K : number of patches to extract, $\mathcal{L}_{\text{task}}$: task specific loss, $\mathcal{I}$: input image, η : parameters of the heatmap network, τ : parameters of the task network.

function TRAINMIST($\mathcal{I}$, $\mathcal{L}_{\text{task}}$)
  for each training batch **do**
    $\tau \leftarrow$ Optimize $\mathcal{T}_\tau$ with $\mathcal{L}_{\text{task}}$
    $\{\mathbf{x}_k^*\} \leftarrow$ Optimize $\{\mathbf{x}_k\}$ with $\mathcal{L}_{\text{task}}$
    $\bar{\mathbf{h}} \leftarrow \mathcal{E}_K^{-1}(\{\mathbf{x}_k^*\})$
    $\eta \leftarrow$ Optimize $\mathcal{H}_\eta$ with $\mathcal{L}_{\text{lift}}$
  end for
end function

4 TRAINING MISTS

The patch selector $\mathcal{E}_K$ identifies the locations of the top-K local maxima of a heatmap and then selects the corresponding patches from the input image. Differentiating this operation provides a gradient with respect to the input, but no gradient with respect to the heatmap. It is thus impossible to train the patch selector parameters directly by backpropagation. Although it is possible to smoothly relax the patch selection operation in the $K = 1$ case (Yi et al., 2016) (i.e. argmax), it is unclear how to generalize this approach to the case of *multiple* distinct local maxima. We thus propose an alternative approach to training our model, using a multi-stage optimization process. Empirically, this optimization process converges smoothly, as we show in Section C of the supplementary material.

Differentiable top-K via lifting. The introduction of auxiliary variables (i.e. lifting) to simplify the structure of an optimization problem has proven effective in a range of domains ranging from non-rigid registration (Taylor et al., 2016), to efficient deformation models (Sorkine & Alexa, 2007), and robust optimization (Zach & Bournaoud, 2018). To simplify our training optimization, we start by decoupling the heatmap tensor from the optimization (1) by introducing the corresponding auxiliary variables $\mathbf{h}$, as well as the patch parameterization variables $\{\mathbf{x}_k\}$ that are extracted by the top-K extractor:

$$\underset{\eta,\tau,\bar{\mathbf{h}},\{\mathbf{x}_k\}}{\operatorname{argmin}} \quad \mathcal{L}_{\text{task}}(\mathcal{T}_\tau(\mathcal{S}(\{\mathbf{x}_k\}))) \quad \text{s.t.} \quad \{\mathbf{x}_k\} = \mathcal{E}_K(\bar{\mathbf{h}}), \quad \bar{\mathbf{h}} = \mathcal{H}_\eta(\mathcal{I}) \tag{4}$$

We then relax (4) to a least-squares penalty:

$$\underset{\eta,\tau,\bar{\mathbf{h}},\{\mathbf{x}_k\}}{\operatorname{argmin}} \quad \mathcal{L}_{\text{task}}(\mathcal{T}_\tau(\mathcal{S}(\{\mathbf{x}_k\}))) + \|\bar{\mathbf{h}} - \mathcal{H}_\eta(\mathcal{I})\|_2^2 \quad \text{s.t.} \quad \{\mathbf{x}_k\} = \mathcal{E}_K(\bar{\mathbf{h}}) \tag{5}$$

and finally approach it by alternating optimization:

$$\underset{\tau,\{\mathbf{x}_k\}}{\operatorname{argmin}} \ \mathcal{L}_{\text{task}}(\mathcal{T}_\tau(\mathcal{S}(\{\mathbf{x}_k\}))) \tag{6}$$

$$\underset{\eta}{\operatorname{argmin}} \ \|\bar{\mathbf{h}} - \mathcal{H}_\eta(\mathcal{I})\|_2^2 \tag{7}$$

where $\bar{\mathbf{h}}$ has been dropped as it is not a free parameter: it can be computed as $\bar{\mathbf{h}} = \mathcal{E}_K^{-1}(\{\mathbf{x}_k\})$ after the $\{\mathbf{x}_k\}$ have been optimized by (6), and as $\bar{\mathbf{h}} = \mathcal{H}_\eta(\mathcal{I})$ after η have been optimized by (7). To accelerate training, we further split (6) into two stages, and alternate between optimizing τ and $\{\mathbf{x}_k\}$. The summary for the three stage optimization procedure is outlined in Algorithm 1: ① we optimize the parameters τ with the loss $\mathcal{L}_{\text{task}}$; ② we then fix τ, and refine the positions of the patches $\{\mathbf{x}_k\}$ with $\mathcal{L}_{task}$. ③ with the optimized patch positions $\{\mathbf{x}_k^*\}$, we invert the top-K operation by creating a target heatmap $\bar{\mathbf{h}}$, and optimize the parameters η of our heatmap network $\mathcal{H}$ using squared ℓ_2 distance between the two heatmaps, $\mathcal{L}_{\text{lift}} = \|\bar{\mathbf{h}} - \mathcal{H}_\eta(\mathcal{I})\|_2^2$. Notice that we are not introducing *any* additional supervision signal that is tangent to the given task.

Generating the target heatmap – $\mathcal{E}_K^{-1}(\{\mathbf{x}_k\})$. For creating the target heatmap $\bar{\mathbf{h}}$, we create a tensor that has zeros everywhere except for the positions corresponding to the optimized positions. However, as the optimized patch parameters are no longer integer values, we need to quantize them with care. For the spatial locations we simply round to the nearest pixel, which at most creates a quantization error of half a pixel, which does not cause problems in practice. For scale however,

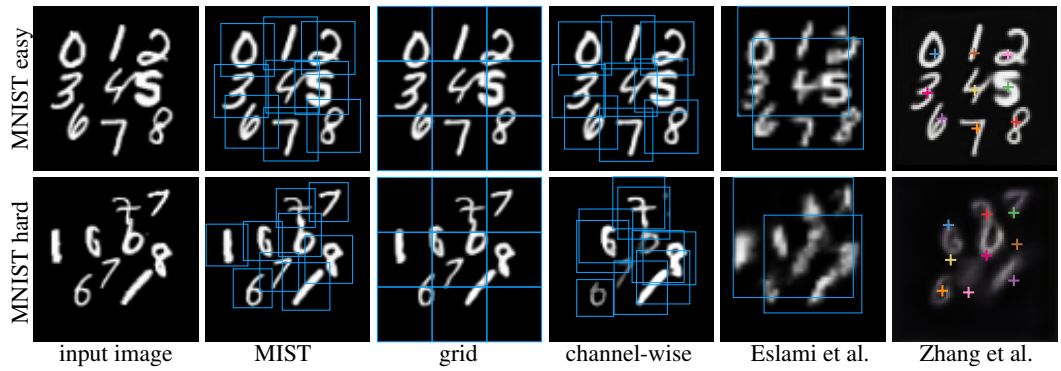

Figure 2: MNIST character synthesis examples for (top) the "easy" single instance setup and (bottom) the hard multi-instance setup. We compare the output of MISTs to grid, channel-wise, sequential Eslami *et al*. (Eslami et al., 2015) and Zhang *et al*. (Zhang et al., 2018c).

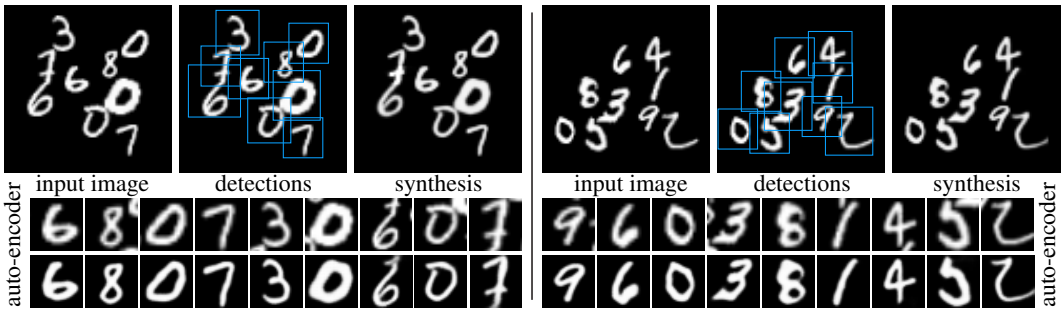

Figure 3: Two auto-encoding examples learnt from MNIST-hard. In the top row, for each example we visualize input, patch detections, and synthesis. In the bottom row we visualize each of the extracted patch, and how it is modified by the learnt auto-encoder.

simple nearest-neighbor assignment causes too much quantization error as our scale-space is sparsely sampled. We therefore assign values to the two nearest neighboring scales in a way that the center of mass would be the optimized scale value. That is, we create a heatmap tensor that would result in the optimized patch locations when used in forward inference.

5 Results and Evaluation

To demonstrate the effectiveness of our framework we evaluate two different tasks. We first perform a quasi-unsupervised image reconstruction task, where *only* the total number of instances in the scene is provided. We then show that our method can also be applied to weakly supervised multi-instance classification, where only image-level supervision is provided. Note that, unlike region proposal based methods, our localization network only relies on cues from the classifier, and *both* networks are trained from scratch.

5.1 Image Reconstruction

From the MNIST dataset, we derive two different scenarios. In the *MNIST easy* dataset, we consider a simple setup where the *sorted* digits are confined to a perturbed *grid* layout; see Figure 2 (top). Specifically, we perturb the digits with a Gaussian noise centered at each grid center, with a standard deviation that is equal to one-eighths of the grid width/height. In the *MNIST hard* dataset, the positions are randomized through a Poisson distribution (Bridson, 2007), as is the identity, and cardinality of each digit. We allow multiple instances of the same digit to appear in this variant. For both datasets, we construct both training and test sets, and the test set is never seen during training.

Comparison baselines We compare our method against four baselines ① the *grid* method divides the image into a 3×3 grid and applies the same auto-encoder architecture as MIST to each grid

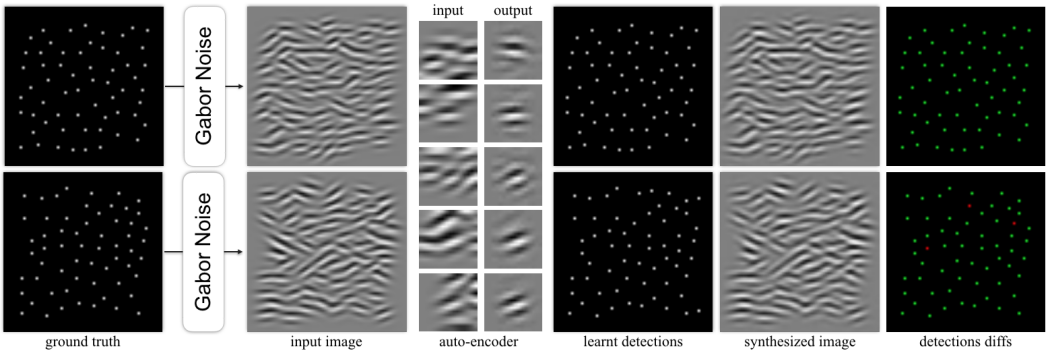

Figure 4: Inverse rendering of Gabor noise; we annotate correct / erroneous localizations.

location to reconstruct the input image; ② the *channel-wise* method uses the same auto-encoder network as MIST, but we modify the heatmap network to produce K channels as output, where each channel is dedicated to an interest point. Locations are obtained through a channel-wise soft-argmax as in Zhang et al. (2018c); ③ *Esl16* (Eslami et al., 2015) is a sequential generative model; ④ *Zha18* (Zhang et al., 2018c) is a state-of-the-art heatmap-based method with channel-wise strategy for unsupervised learning of landmarks. For more details see Supplementary Section B.

Results for "MNIST easy" As shown in Figure 2 (top) all methods successfully re-synthesize the image, with the exception of Eslami et al. (Eslami et al., 2015). As this method is sequential, with nine digits the sequential implementation simply becomes too difficult to optimize through. Note how this method only learns to describe the scene with a few large regions. We summarize quantitative results in Table 1.

Results for "MNIST hard" As shown in Figure 2 (bottom), all baseline methods failed to properly represent the image. Only MIST succeeded at both localizing digits and reconstructing the original image. Although the grid method accurately reconstructs the image, it has no concept of individual digits. Conversely, as shown in Figure 3, our method generates accurate bounding boxes for digits even when these digits overlap, and does so without any location supervision. For quantitative results, please see Table 1.

Finding the basis of a procedural texture We further demonstrate that our methods can be used to find the basis function of a procedural texture. For this experiment we synthesize textures with procedural Gabor noise (Lagae et al., 2009). Gabor noise is obtained by convolving oriented Gabor wavelets with a Poisson impulse process. Hence, given exemplars of noise, our framework is tasked to regress the underlying impulse process and reconstruct the Gabor kernels so that when the two are convolved, we can reconstruct the original image. Figure 4 illustrates the results of our experiment. The auto-encoder learned to accurately reconstruct the Gabor kernels, even though in the training images they are heavily overlapped. These results show that MIST is capable of generating and reconstructing large numbers of instances per image, which is *intractable* with other approaches.

5.2 Multiple instance classification

Multi-MNIST – Figure 5. To test our method in a multiple instance classification setup, we rely on the *MNIST hard* dataset. We compare our method to *channel-wise*, as other baselines are designed for generative tasks. To evaluate the detection accuracy of the models, we compute the intersection over union (IoU) between the ground-truth bounding box and the detection results, and assign it

	MIST	Grid	Ch.-wise	Esl16	Zha18
MNIST easy	**.038**	.039	.042	.100	.169
MNIST hard	**.053**	.062	.128	.154	.191
Gabor	**.095**	-	-	-	-

Table 1: Reconstruction error (root mean square error). Note that Grid *does not* learn any notion of digits.

	MIST	Ch.-wise	Supervised
IOU 50%	**97.8%**	25.4%	99.6%
Classif.	**98.8%**	75.5%	98.7%
Both	**97.5%**	24.8%	98.6%

Table 2: Instance level detection and classification performance on "MNIST hard".

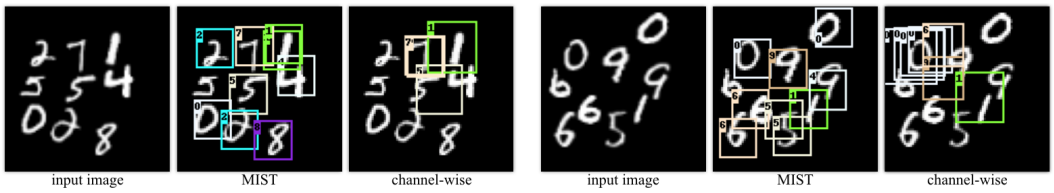

Figure 5: Two qualitative examples for detection and classification on our Multi-MNIST dataset.

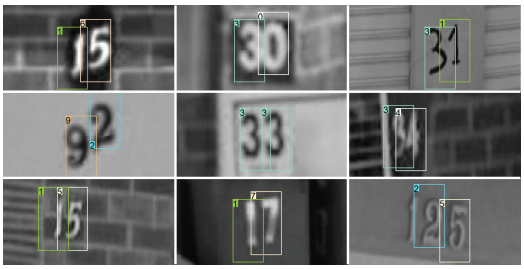

Figure 6: Qualitative SVHN results.

	MIST	Supervised
$AP^{IoU=.00}$	**82.6%**	65.6%
$AP^{IoU=.50}$	**76.5%**	62.8%
$AP^{IoU=.60}$	**63.7%**	59.8%
$AP^{IoU=.70}$	42.7%	**51.9%**
$AP^{IoU=.80}$	19.9%	**34.6%**
$AP^{IoU=.90}$	4.2%	**11.0%**

Table 3: Quantitative SVHN results.

as a match if the IoU score is over 50%. We report the number of correctly classified matches in Table 2, as well as the proportion of instances that are both correctly detected and correctly classified. Our method clearly outperforms the *channel-wise* strategy. Note that, even without localization supervision, our method correctly localizes digits. Conversely, the *channel-wise* strategy fails to learn. This is because *multiple instances* of the same digits are present in the image. For example, in the example2 Figure 5 (right), we have two number sizes, zeros, and nines. This prevents any of these digits from being detected/classified properly by a channel-wise approach.

SVHN – Figure 6 and Table 3. We further apply MIST to the uncropped and unaligned Street View House Numbers dataset (Netzer et al., 2011). Compared to previous work that has used cropped and resized SVHN images (*e.g.* (Netzer et al., 2011; Ba et al., 2015; Goodfellow et al., 2014; Jaderberg et al., 2015)), this evaluating setting is significantly more challenging, because digits can appear anywhere in the image. We resize all images to 60×240, use only images labeled as containing 2 digits, and apply MIST at a single scale. Although the dataset provides bounding boxes for the digits, we ignore these bounding boxes and use only digit labels as supervision. During testing, we exclude images with small bounding boxes (< 30 pixels in height). We report results in terms of $AP^{IoU=.X}$, where X is the threshold for determining detection correctness. With $IoU = 0$, we refer to a "pure" classification task (i.e. no localization). As shown, supervised results provide better performance with higher thresholds, but MIST performs even better than the supervised baseline for moderate thresholds. We attribute this to the fact that, by providing direct supervision on the location, the training focuses too much on having high localization accuracy.

6 Conclusion

In this paper, we introduce the MIST framework for multi-instance image reconstruction/classification. Both these tasks are based on localized analysis of the image, yet we train the network without providing any localization supervision. The network learns how to extract patches on its own, and these patches are then fed to a task-specific network to realize an end goal. While at first glance the MIST framework might appear non-differentiable, we show how via lifting they can be effectively trained in an end-to-end fashion. We demonstrated the effectiveness of MIST by introducing a variant of the MNIST dataset, and demonstrating compelling performance in both reconstruction and classification. We also show how the network can be trained to reverse engineer a procedural texture synthesis process. MISTs are a first step towards the definition of optimizable image-decomposition networks that could be extended to a number of exciting *unsupervised* learning tasks. Amongst these, we intend to explore the applicability of MISTs to unsupervised detection/localization of objects, facial landmarks, and local feature learning.

REFERENCES

J. L. Ba, V. Mnih, and K. Kavukcuoglu. Multiple Object Recognition With Visual Attention. In *ICLR*, 2015.

H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. *CVIU*, 10(3): 346–359, 2008.

R. Bridson. Fast Poisson Disk Sampling in Arbitrary Dimensions. In *SIGGRAPH sketches*, 2007.

Y-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar. Rethinking the Faster R-CNN Architecture for Temporal Action Localization. In *CVPR*, 2018.

D. Detone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-Supervised Interest Point Detection and Description. *CVPR Workshop on Deep Learning for Visual SLAM*, 2018.

P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian Detection: An Evaluation of the State of the Art. *PAMI*, 34(4):743–761, 2012.

S. M. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, K. Kavukcuoglu, and G. E. Hinton. Attend, Infer, Repeat: Fast Scene Understating with Generative Models. In *NeurIPS*, 2015.

P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *PAMI*, 32(9):1627–1645, 2010.

M. Gao, A. Li, V. I. Morariu, and L. S. Davis. C-WSL: Coung-guided Weakly Supervised Localization. In *ECCV*, 2018.

G. Georgakis, S. Karanam, Z. Yu, J. Ernst, and J. Košecká. End-to-end Learning of Keypoint Detector and Descriptor for Pose Invariant 3D Matching. In *CVPR*, 2018.

I. Goodfellow, Y. Bularov, J. Ibarz, S. Arnoud, and V. Shet. Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks. In *ICLR*, 2014.

K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra. DRAW: A Recurrent Neural Network For Image Generation. In *ICML*, 2015.

A. Gupta, A. Vedaldi, and A. Zisserman. Inductive Visual Localization: Factorised Training for Superior Generalization. In *BMVC*, 2018.

K. He, X. Zhang, R. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification. In *ICCV*, 2015.

K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.

N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa. Cross-Domain Weakly-Supervised Object Detection through Progressive Domain Adaptation. In *ECCV*, 2018.

M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial Transformer Networks. In *NeurIPS*, pp. 2017–2025, 2015.

Wei Jiang, Weiwei Sun, Andrea Tagliasacchi, Eduard Trulls, and Kwang Moo Yi. Linearized multi-sampling for differentiable image transformation. *arXiv Preprint*, 2019. URL `http://arxiv.org/abs/1901.07124`.

J. Johnson, A. Karpathy, and L. Fei-fei. Densecap: Fully Convolutional Localization Networks for Dense Captioning. In *CVPR*, 2016.

J. Johnson, A. Gupta, and L. Fei-fei. Image Generation from Scene Graphs. In *CVPR*, 2018.

Ares Lagae, Sylvain Lefebvre, George Drettakis, and Philip Dutré. Procedural noise using sparse gabor convolution. *TOG*, 2009.

D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 20(2), 2004.

X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley. Least Squares Aenerative Adversarial Networks. In *CVPR*, 2017.

D. Merget, M. Rock, and G. Rigoll. Robust Facial Landmark Detection via a Fully-Conlolutional Local-Global Context Network. In *CVPR*, 2018.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *Deep Learning and Unsupervised Feature Learning Workshop, NeurIPS*, 2011.

A. Newell, K. Yang, and J. Deng. Stacked Hourglass Networks for Human Pose Estimation. In *ECCV*, 2016.

Y. Ono, E. Trulls, P. Fua, and K. M. Yi. Lf-Net: Learning Local Features from Images. In *NeurIPS*, 2018.

J. Redmon and A. Farhadi. YOLO 9000: Better, Faster, Stronger. In *CVPR*, 2017.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*, 2016.

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In *ICCV*, 2017.

R. Sewart and M. Andriluka. End-to-End People Detection in Crowded Scenes. In *CVPR*, 2016.

Yunhan Shen, Rongrong Ji, Shengchuan Zhang, Wangmeng Zuo, and Yan Wang. Generative adversarial learning towards fast weakly supervised detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5764–5773, 2018.

B. Singh, H. Li, A. Sharma, and L. S. Davis. R-FCN-3000 at 30fps: Decoupling Detection and Classification. In *CVPR*, 2018.

Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *SGP*, 2007.

M. Sun, Y. Yuan, F. Zhou, and E. Ding. Multi-Attention Multi-Class Constraint for Fine-grained Image Recognition. In *ECCV*, 2018.

S. Suwajanakorn, N. Snavely, J. Tompson, and M. Norouzi. Discovery of Latent 3D Keypoints via End-To-End Geometric Reasoning. In *NeurIPS*, 2018.

P. Tang, X. Wang, A. Wang, Y. Yan, W. Liu, J. Huang, and A. Yuille. Weakly Supervised Region Proposal Network and Object Detection. In *ECCV*, 2018.

Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, Arran Topalian, Erroll Wood, Sameh Khamis, Pushmeet Kohli, Toby Sharp, Shahram Izadi, Richard Banks, Andrew Fitzgibbon, and Jamie Shotton. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *TOG*, 2016.

B. Tekin, P. Marquez-neila, M. Salzmann, and P. Fua. Learning to Fuse 2D and 3D Image Cues for Monocular Body Pose Estimation. In *ICCV*, 2017.

J. R. R Uijlings, S. Popov, and V. Ferrari. Revisiting Knowledge Transfer for Training Object Class Detectors. In *CVPR*, 2018.

F. Wan, P. Wei, J. Jiao, Z. Han, and Q. Ye. Min-Entropy Latent Model for Weakly Supervised Object Detection. In *CVPR*, 2018.

F. Wang, L. Zhao, X. Li, X. Wang, and D. Tao. Geometry-Aware Scene Text Detection with Instance Transformation Network. In *CVPR*, 2018.

K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned Invariant Feature Transform. In *ECCV*, 2016.

C. Zach and G. Bournaoud. Descending, Lifting or Smoothing: Secrets of Robust Cost Opimization. In *ECCV*, 2018.

S. Zhang, J. Yang, and B. Schiele. Occluded Pedestrian Detection Through Guided Attention in CNNs. In *CVPR*, 2018a.

X. Zhang, Y. Wei, G. Kang, Y. Wang, and T. Huang. Self-produced Guidance for Weakly-supervised Object Localization. In *ECCV*, 2018b.

Y. Zhang, Y. Gui, Y. Jin, Y. Luo, Z. He, and H. Lee. Unsupervised Discovery of Object Landmarks as Structural Representations. In *CVPR*, 2018c.

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. In *CVPR*, 2016.

Appendix

A Comparison to LF-Net

Note that differently from LF-Net (Ono et al., 2018), we do not perform a softmax along the scale dimension. The scale-wise softmax in LF-Net is problematic as the computation for a softmax function relies on the input to the softmax being *unbounded*. For example, in order for the softmax function to behave as a max function, due to exponentiation, it is necessary that one of the input value reaches infinity (i.e. the value that will correspond to the max), or that all other values to reach negative infinity. However, at the network stage where softmax is applied in (Ono et al., 2018), the score range from zero to one, effectively making the softmax behave similarly to averaging. Our formulation does not suffer from this drawback.

B Implementation details

MIST auto-encoder network. The input layer of the autoencoder is $32 \times 32 \times C$ where C is the number of color channels. We use 5 up/down-sampling levels. Each level is made of 3 standard non-bottleneck ResNet v1 blocks (He et al., 2015) and each ResNet block uses a number of channels that doubles after each downsampling step. ResNet blocks uses 3×3 convolutions of stride 1 with ReLU activation. For downsampling we use 2D max pooling with 2×2 stride and kernel. For upsampling we use 2D transposed convolutions with 2×2 stride and kernel. The output layer uses a sigmoid function, and we use layer normalization before each convolution layer.

MIST classification network. We re-use the same architecture as encoder for first the task and append a dense layer to map the latent space to the score vector of our 10 digit classes.

Baseline unsupervised reconstruction methods. To implement the *Eslami* et al. *(Eslami et al., 2015)* baseline, we use a publicly available reimplementation.[1] We note that Eslami *et al*. (Eslami et al., 2015) originally applied their model to a dataset consisting of images of 0, 1, or 2 digits with equal probability. We found that the model failed to converge unless it was trained with examples where various number of total digits exist, so for fair comparison, we populate the training set with images consisting of all numbers of digits between 0 and 9. For the *Zhang* et al. *(Zhang et al., 2018c)* baseline, we use the authors' implementation and hyperparameters.

C Convergence during training

As is typical for neural network training, our objective is non-convex and there is no guarantee that a local minimum found by gradient descent training is a global minimum. Empirically, however, the optimization process is stable, as shown in Figure 7. Early in training, keypoints are detected at *random* locations as the heatmaps are generated by networks with randomly initialized weights. However, as training continues, keypoints that, by chance, land on locations nearby the correct object (e.g. numbers) for certain samples in the random batch, and become reinforced. Thus, ultimately MIST learns to detect these locations and perform the task of interest. Note that this is unsurprising, as our formulation is a lifted version of this loss to allow gradient-based training. Figure 7(right) also shows the evolution of the heatmap starting from a random-like signal (top) and converging to a highly peaked response (bottom).

D Non-uniform distributions

Although the images we show in Figure 2 involve small displacements from a uniformly spaced grid, our method does not require the keypoints to be evenly spread. As shown in Figure 8, our method is able to successfully learn even when the digits are placed unevenly. Note that, as our detector is fully convolutional and local, it cannot learn the absolute location of keypoints. In fact, we weakened the randomness of the locations for fairness against (Zhang et al., 2018c), which is not designed to deal with severe displacements.

[1] `https://github.com/aakhundov/tf-attend-infer-repeat`

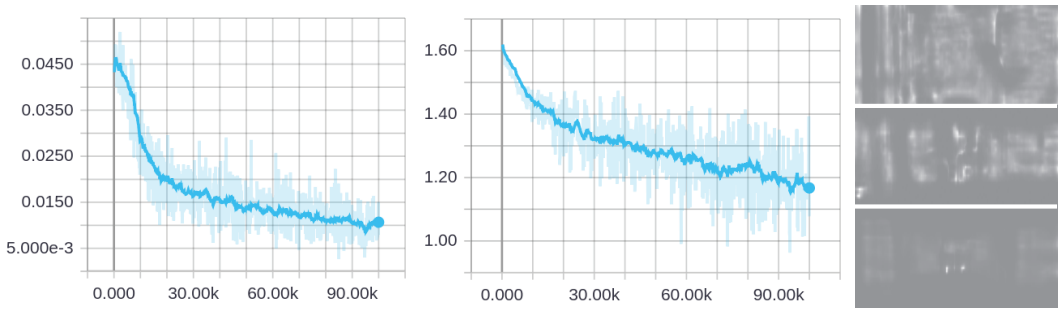

Figure 7: Evolution of the loss during training. (left) The classification loss. (middle) The heatmap loss. (right) The heatmap evolution over training iterations (from top to bottom) on an SVHN example image.

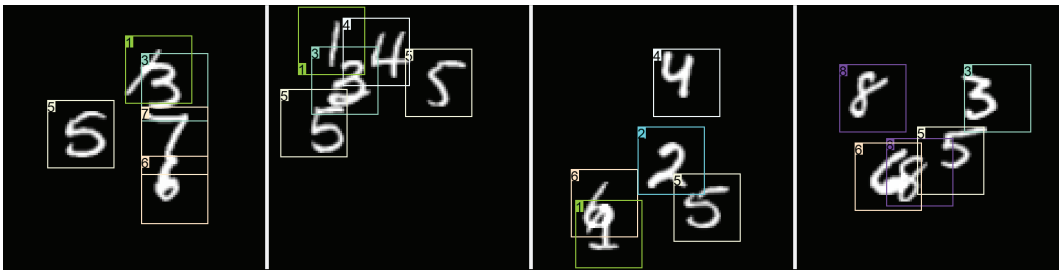

Figure 8: Examples with uneven distributions of digits.

E VARYING NUMBER OF OBJECTS

We currently require that we know the number of objects in a scene. Nonetheless, our additional study in Table 4 reveals that our method has tolerance towards this number. Specifically, we train with a fixed number of K, with varying number of ground-truth number of digits. For example, with $K = 6$, where the ground truth could be anything within $\{3, 4, 5, 6, 7, 8, 9\}$. We then evaluate how the framework performs at inference, given ground-truth K. Our method still is able to give accurate result with inaccurate K during training.

Furthermore, knowing the exact number of objects is not a strict requirement at test time, as our detector generates a heatmap for the entire image regardless of the K it was trained with. Selecting the K that is most appropriate to a given heatmap is still an ongoing work, but using a simple threshold should yield reasonable results. Note also that while in theory sequential methods are free from this limitation, in practice they are able to deal with limited number of objects (e.g. up to three) due to their recurrent nature. They also often generalize poorly to numbers of objects different than those observed during training.

K_{train}	$AP^{IoU=.50}$
$\{\mathbf{9}\}$	**92.2%**
$\{\mathbf{8}, 9\}$	90.2%
$\{7, \mathbf{8}, 9\}$	90.7%
$\{6, \mathbf{7}, 8, 9\}$	90.1%
$\{5, 6, \mathbf{7}, 8, 9\}$	87.4%
$\{4, 5, \mathbf{6}, 7, 8, 9\}$	88.0%
$\{3, 4, 5, \mathbf{6}, 7, 8, 9\}$	90.8%
Supervised	**98.6%**

Table 4: Sensitivity experiment of K on the MNIST hard dataset. K_{train} indicates the number of digits in the training set. Bold K numbers indicate the fixed number of keypoints used by the network at training. For fair comparison, the exact K is used at test time.