GENERATING UNOBSERVED ALTERNATIVES: A CASE STUDY THROUGH SUPER-RESOLUTION AND DECOMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

We consider problems where multiple predictions can be considered correct, but only one of them is given as supervision. This setting differs from both the regression and class-conditional generative modelling settings: in the former, there is a unique observed output for each input, which is provided as supervision; in the latter, there are many observed outputs for each input, and many are provided as supervision. Applying either regression methods and conditional generative models to the present setting often results in a model that can only make a single prediction for each input. We explore several problems that have this property, which naturally arise in image processing, and develop an approach that can generate multiple high-quality predictions given the same input. As a result, it can be used to generate high-quality outputs that are different from the observed output.

1 INTRODUCTION

Supervised learning is centred around prediction. In the classification or regression setting, only a single label/target is assumed to be correct, and the goal is predict the label with high confidence or generate a prediction that is as close as possible to the target. In settings such as multi-label prediction or class-conditional generative modelling, there could be *multiple* prediction targets for the same input that are all correct. For example, in class-conditional generative modelling, the input is the class label and all data points that belong to that class are correct prediction targets. Multiple prediction targets for the same input are given as supervision, and the goal is to generate *all* such prediction targets for the same input (class label).

In this paper, we consider a different problem setting with the following properties: (1) for the same input, there could be *multiple* prediction targets that are correct, but (2) only a single prediction target per input is given as supervision. The goal is still to generate all prediction targets for the same input. See Table 1 for a comparison of the problem setting we consider to other common settings. Note that we focus on the case of continuous prediction targets and leave discrete labels to future work.

Problem Setting	Label Type	Prediction	Supervision
Regression Classification	Continuous Discrete	One-to-one	One-to-one
Class-conditional Generative Modelling Multi-label Prediction	Continuous Discrete	One-to-many	One-to-many
Present Setting	Continuous	One-to-many	One-to-one

Table 1: Comparison of the problem setting we consider to other common settings.

When do such prediction problems arise? They often come up in inverse problems, which require generating *more* information from *less* information, information that is not present in the input. For example, consider the problem of super-resolution, which aims to generate a high-resolution

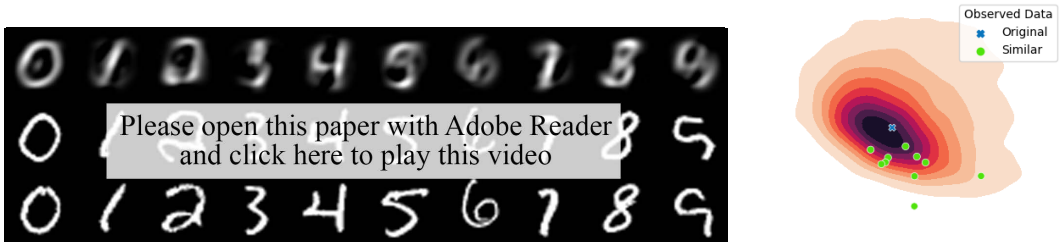
image from a low-resolution image. The high-frequency details are completely missing from the low-resolution image, but they must be generated in the high-resolution image.

Inverse problems are typically *ill-posed*, that is, the input cannot uniquely determine the output and so there could be multiple valid outputs for the same input. However, only one of them is actually observed. Concretely, in the case of super-resolution, there are many ways to generate details in the high-resolution image, and the observed high-resolution image used for training represents only one of these ways. This combination of *one-to-many* prediction and *one-to-one* supervision characterizes the problem setting we consider.

The problem essentially requires us to generate alternatives that were never observed, so a natural question is why it should be possible at all. After all, if there were a valid alternative output that was never realized, how do we know whether it exists, and why should the model generate such an alternative if there is no indication that it exists? The answer lies in an observation that holds true across many natural problems: *which* of the many valid prediction targets is observed is usually arbitrary, and so while a valid alternative for the current input may not be observed, we expect an analogous version of it for *some* other input to be observed. Therefore, the hope is for the model to generalize across different inputs to produce the full range of alternative predictions for all inputs.

In this paper, we take an initial step towards addressing this problem and propose an approach for it based on Implicit Maximum Likelihood Estimation (IMLE) (Li & Malik, 2018). We demonstrate on three problems that the approach can produce different alternative predictions for the same input, even though only one prediction target is given for each input.

2 An Illustrative Example using MNIST



(a) Samples (b) KDE

Figure 1: Example unseen input digits and outputs from our method. Top row is the input, middle row is the predictions and bottom row is the observed output.

To illustrate the problem setting, we will start with a simple illustrative example using MNIST. We consider the problem of predicting from the first ten principal components of a data point the values of the remaining ones. More concretely, we perform principal component analysis (PCA) and project each data point onto the PCA basis. The input is the image reconstructed from the first ten coordinates and the observed output is the original image.

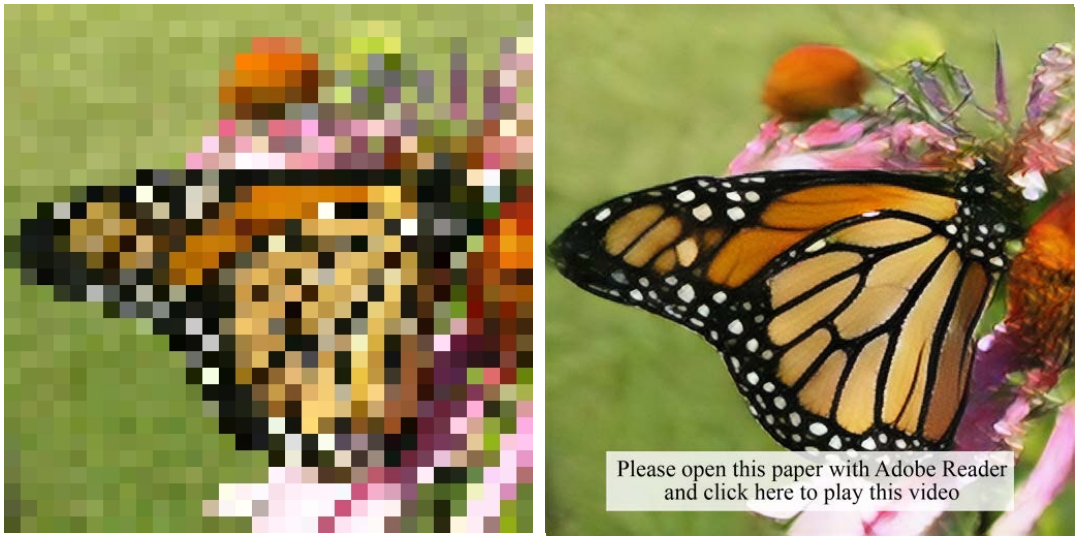
This prediction problem is inherently one-to-many, but only one-to-one supervision is available. Specifically, given the first ten coordinates of a real data point, there are many possible ways to fill in the values of the remaining coordinates that will result in plausible MNIST digits. However, only one of these is observed, namely the original real data point.

To illustrate what the unobserved alternatives could be, we visualize the results of our method (the details of which will be discussed later) in Figure 1a. All the predictions share the same first ten coordinates, but differ in the remaining ones. As shown, all predictions are plausible, but differ from the observed output.

We can visualize the marginal distribution over the 11th and 12th coordinates of the predictions and compare to those of the real data point. As shown in Figure 1b, the real data point lies in a high density region of the prediction distribution, suggesting the method is able to predict the real data point (or at least the 11th and 12th coordinates). Note that there is only a *single* data point we can

observe for the given input, because other data points in the dataset have different coordinates along the first 10 principal components and therefore differ from the given input.

As a proxy for other data points that *could* have been observed for the given input, we visualize ten data points whose first 10 principal components are the *closest* to the given input. While they technically do not match the given input (because the first 10 principal components are different from the given input), they are hopefully similar to unobserved alternatives and can therefore give us a sense of how the unobserved alternatives would be distributed. As shown, the prediction distribution has moderately high density at most of these points, indicating that they can be predicted by the method.



(a) Input (b) Output

Figure 2: Example unseen input image and output from our method (HyperRIM). Click on (b) to see output of model while it trains, demonstrating stable training.

3 METHOD

One-to-many prediction problems can be naturally formulated in probabilistic terms. If we use $\mathbf{x}$ to denote the input, $\mathbf{y}$ to denote the prediction, our goal is to learn $p(\mathbf{y}|\mathbf{x})$. Ideally $p(\mathbf{y}|\mathbf{x})$ should assign high probability density to both observed and unobserved valid predictions, and low probability density elsewhere. So, each mode of $p(\mathbf{y}|\mathbf{x})$ corresponds to a valid prediction.

Regression models take the form of a deterministic function from $\mathbf{x}$ to $\mathbf{y}$, and so $p(\mathbf{y}|\mathbf{x})$ is always a delta. In order to produce non-deterministic predictions, the most direct way to extend regression models is to add a latent random variable as an input to the deterministic function. More precisely, a prediction is given by $\mathbf{y} \coloneqq T_\theta(\mathbf{x}, \mathbf{z})$ where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$. This is variously known as an implicit generative model (Mohamed & Lakshminarayanan, 2016), a neural sampler (Nowozin et al., 2016) or a decoder-based model (Wu et al., 2016).

Such a model can be trained as a conditional GAN, where $T_\theta(\cdot, \cdot)$ is interpreted as the generator. In practice, due to mode collapse, some valid predictions cannot be produced by the generator. This problem is exacerbated in the presently considered setting with one-to-one supervision: since there is only one observed output $\mathbf{y}$ for each input $\mathbf{x}$, there is only one mode to collapse to. As a result, all samples of the generator conditioned on thee same input $\mathbf{x}$ are identical and the random variable $\mathbf{z}$ is effectively ignored. Hence, the generator becomes a deterministic mapping from $\mathbf{x}$ to $\mathbf{y}$, akin to a vanilla regression model.

To obtain non-determinsitic predictions $\mathbf{y}$ despite the availability of only a single observation, we propose training the model using Implicit Maximum Likelihood Estimation (IMLE), which avoids mode collapse.

3.1 IMPLICIT MAXIMUM LIKELIHOOD ESTIMATION (IMLE)

Implicit Maximum Likelihood Estimation (IMLE) (Li & Malik, 2018) is a method for training implicit generative models. Compared to GANs, there are two differences: it explicitly aims to cover all modes, and optimizes a non-adversarial objective. To achieve the former, IMLE reverses the direction in which generated samples are matched to real data: rather than making each generated sample similar to some real data point, it makes sure each real data point has a similar generated sample. To achieve the latter, it removes the discriminator (which matches generated samples to real data implicitly) and instead explicitly performs matching using nearest neighbour search. The latter can be done efficiently using DCI (Li & Malik, 2016; 2017), which avoids the curse of dimensionality.

More precisely, if we denote the generator parameterized by θ as $T_\theta(\cdot)$, which takes in a random code $\mathbf{z}_j$ and outputs a sample, IMLE optimizes the following objective:

$$\min_\theta \mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_n \sim \mathcal{N}(0,\mathbf{I})} \left[\sum_{i=1}^{n} \min_{j \in \{1,\ldots,m\}} d(T_\theta(\mathbf{z}_j), \mathbf{y}_i) \right],$$

where $\mathbf{y}_i$ is a real data point, $d(\cdot,\cdot)$ is a distance metric and m is a hyperparameter.

3.2 CONDITIONAL IMLE

IMLE can be extended to model conditional distributions by separately applying IMLE to each member of a family of distributions $\{p(\mathbf{y}|\mathbf{x}_i)\}_{i=1}^n$. If we denote the generator as $T_\theta(\cdot,\cdot)$, which takes in an input $\mathbf{x}_i$ and a random code $\mathbf{z}_{i,j}$ and outputs a sample from $p(\cdot|\mathbf{x}_i)$, the method optimizes the following objective:

$$\min_\theta \mathbb{E}_{\mathbf{z}_{1,1},\ldots,\mathbf{z}_{n,m} \sim \mathcal{N}(0,\mathbf{I})} \left[\sum_{i=1}^{n} \min_{j \in \{1,\ldots,m\}} d(T_\theta(\mathbf{x}_i, \mathbf{z}_{i,j}), \mathbf{y}_i) \right],$$

where $\mathbf{y}_i$ is the observed output that corresponds to $\mathbf{x}_i$, $d(\cdot,\cdot)$ is a distance metric and m is a hyperparameter. We use LPIPS perceptual distance (Zhang et al., 2018) as our distance metric.

3.3 MODEL ARCHITECTURE

Different types of generative models require different architectures due to differences in behaviour (e.g.: mode seeking vs. covering) and training dynamics (e.g.: adversarial vs. non-adversarial) (Van den Oord et al., 2016; Vahdat & Kautz, 2020; Radford et al., 2015). In this paper, we introduce a new architecture for IMLE which substantially outperforms prior IMLE architectures (Li & Malik, 2018; Li et al., 2020). As we will show later, this is critical to generating high quality images.

The model architecture relies on a backbone consisting of two branches. The first branch mainly consists of a sequence of residual-in-residual dense blocks (RRDB) (Wang et al., 2018a), which is a sequence of three dense blocks (Fig. 4b) connected by residual connections (Fig. 4a). The second branch consists of a mapping network Karras et al. (2019) produces a scaling factor and an offset for each of the feature channels after each RRDB in the first branch. Additionally we added weight normalization (Salimans & Kingma, 2016) to all convolution layers. While various design motifs are inspired by other works, combining them in a way that gave good performance when trained with IMLE was non-trivial and required thorough experimentation. We found the optimal hyperparameter settings to differ substantially between GAN-based and IMLE-based architectures. For example, we reduced the number of RRDB blocks by a factor of 4 and substantially expanded the number of channels compared to ESRGAN. What is new is not the design motifs themselves, but the development of an architecture for IMLE that can generate high-quality images. We expect this to be of practical interest in broader contexts, because this architecture combined with IMLE can offer benefits that cannot be obtained with other methods, such as training stability, mode coverage, fast sampling and high-quality samples.

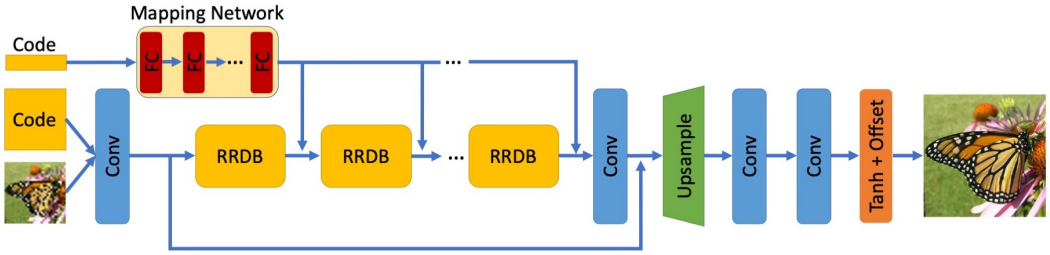
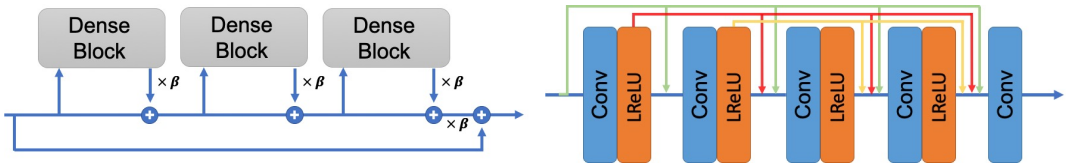


Figure 3: Details of the architecture backbone. See Figure 4a for the inner workings of RRDB blocks.



(a) Residual-in-Residual Dense Block (RRDB)

(b) Dense Block

Figure 4: (a) Inner workings of Residual-in-Residual Dense Blocks (RRDBs), which comprises of dense blocks (details in (b)). β is the residual scaling parameter. (b) Inner workings of dense blocks.

4 Super-Resolution

Single-image super-resolution (SISR) is a classic problem in image processing. Applications span consumer and industrial use cases, and range from photo enhancement to medical imaging. Most methods consider moderately low upscaling factors (e.g. $2-4\times$). We consider an upscaling factor of $16\times$, where the width and height are both increased by 16 times, and so the number of pixels is increased by 256 times. Under this setting, the input contains much less information about the output, and so there could be a lot more valid output images for the same input image. The problem therefore represents an ideal testbed for our method.

4.1 Progressive Upscaling

We adopt an approach of progressively upscaling, where we upscale the image by 2 times at a time. We chain together four backbone architectures which become sub-networks in a larger architecture, as shown in Figure 5. Each sub-network takes a latent code and the output of the previous sub-network, or if there is no previous sub-network, the input image.

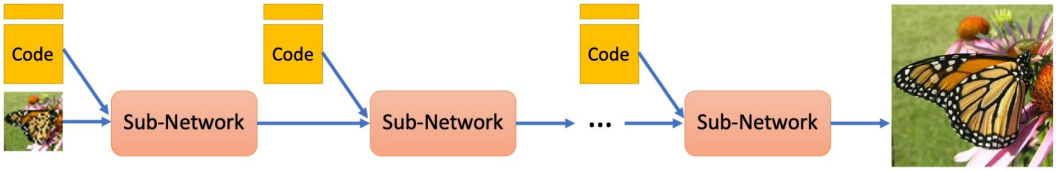


Figure 5: Our HyperRIM model consists of multiple sub-networks, each of which upscales by a factor of 2 and concatenates a random code to its input.

We add intermediate supervision to the output of each sub-network, so that the distance metric in IMLE is chosen to be the sum over LPIPS distances between the output of each sub-network and the original image downsampled to the same resolution.

Additionally, we use a hierarchical sampling procedure to generate the pool of samples IMLE operates over. Because conditional IMLE only uses the sample that is most similar to the observed output for backpropagation, we can improve the sample efficiency by sampling only in the region likely to

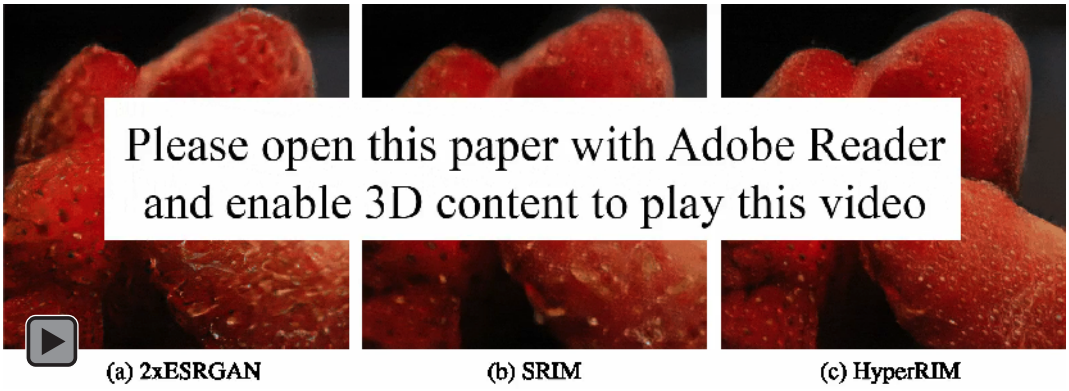


Figure 6: Visualization of different samples generated by our method (HyperRIM) and the baselines. As shown, (a) generates near-identical samples, (b) generates diverse samples that are less visually plausible and less faithful to the colours of the input, (c) generates samples that are both diverse and consistent with the input.

be close to the observed output, which can be viewed as a way of increasing the effective number of samples. To this end, we generate a set of latent codes for the first sub-network and select the latent vector that corresponds to the sub-network output that results in a sample that is most similar to the downsampled observed output. Then for each subsequent sub-network, we fix the latent codes for all previous sub-networks and generate a set of latent codes only for the current sub-network, effectively drawing samples conditioned on the selected latent codes for lower resolutions.

4.2 Experimental Setting

We used a subset of three categories from the ILSVRC-2012 dataset consisting of 3900 images. To obtain the input and target output images, we downsampled them anisotropically to 512×512 and 32×32 respectively using a bilinear filter. The train and test images are disjoint.

We compare our method, HyperRIM, to leading GAN-based and IMLE-based methods, namely ESRGAN (Wang et al., 2018a) and SRIM (Li et al., 2020). ESRGAN is a conditional GAN trained with the relativistic GAN objective (Jolicoeur-Martineau, 2018) and also uses two auxiliary losses on raw pixels and VGG features. Since ESRGAN was originally designed for $4\times$ upscaling, we stack two separate ESRGAN models to upscale the input image by $16\times$. To make the generator capable of producing non-deterministic predictions, we concatenate a random code to the inputs of both models.

4.3 Quantitative Results

We evaluate all methods according to two metrics, Fréchet Inception Distance (FID) (Heusel et al., 2017) and faithfulness-weighted variance (Li et al., 2020). The former measures perceptual quality of the output images, while the latter measures the diversity of the different output images for the same input image weighted by their consistency with the original image.

As shown in Table 7a, HyperRIM outperformed both baselines in terms of FID, indicating that it produces higher-quality images than both. As shown in Table 2, HyperRIM achieved higher faithfulness-weighted variance than the baselines at all but the highest bandwidth parameters. At higher bandwidth parameters, there is a lower penalty on producing outputs that are inconsistent with the original image. So, the outputs of our method are more diverse and consistent with the original image.

4.4 Qualitative Results

We show the results of our method and the baselines in Appendix C . As shown, HyperRIM generates better quality results than all baselines. Figure 6 shows a video of the different outputs for the same input produced by each method. As shown, the outputs generated by HyperRIM are more realistic

	2xESRGAN	SRIM	HyperRIM
FID	21.61	28.85	**19.31**

(a) Super-Resolution

	Pix2Pix	HyperRIM
FID	110.80	**94.84**

(b) Image Decompression

Figure 7: Comparison of Fréchet Inception Distance (FID) to the target of the samples generated by our method (HyperRIM) and the baselines. Lower values of FID are better. We compare favourably on this perceptual metric (FID).

σ	2xESRGAN	SRIM	HyperRIM
0.3	2.83×10^{-2}	$\mathbf{4.90 \times 10^{-2}}$	4.84×10^{-2}
0.2	1.48×10^{-3}	4.09×10^{-3}	$\mathbf{4.25 \times 10^{-3}}$
0.15	5.30×10^{-5}	1.88×10^{-4}	$\mathbf{2.04 \times 10^{-4}}$

Table 2: Comparison of faithfulness weighted variance of the samples generated by our method (HyperRIM) and the baselines. Higher value shows more variation in the generated samples that are faithful to the observed output. σ is the bandwidth parameter for the Gaussian kernel used to compute the faithfulness weights.

and diverse than those generated by the baselines. We also visualize the precision and recall of various methods in Appendix A.

In Figure 2, we visualize the output of HyperRIM for a test input image while training. As shown, the quality of output improves steadily during training, thereby demonstrating training stability.

5 Image Decompression

Most images are stored in a compressed format such as JPEG, and the original uncompressed images are lost. Significant artifacts may result when the images are decompressed using JPEG; to restore the original quality of images that are only stored in compressed form, it would be beneficial to learn to generate the original image from the compressed version. The input does not contain enough information to uniquely determine the output, and so it would be useful to produce multiple plausible uncompressed images and allow the user to choose one to their liking.

We choose a single backbone network as our architecture with one change: we removed the upsampling layer because the input and output resolutions are the same for decompression.

5.1 Experimental Setting

To generate training data, we compressed each image from the RAISE1K (Dang-Nguyen et al., 2015) dataset using JPEG with a quality of 1. We compare our method to Pix2Pix (Isola et al., 2017), given the lack of a dedicated method for image decompression [1].

5.2 Results

We compare the results to the baseline in terms of FID in Table 7b. Our method, HyperRIM, achieves a lower FID than the baseline, demonstrating better perceptual quality. We visualize the outputs of our method and Pix2Pix in Figure 8. As shown, our method was able to remove most blocky artifacts, including those on the face and shoulder of the statue. Additionally, our method can recover different output images with different colour tones. This makes sense, because JPEG compression can cause global colour distortions.

[1] Not to be confused with learned image compression methods, which changes the way the image is encoded. In this setting, we are given the JPEG encoded image, and so compression methods cannot be used.

(a) Input (b) Pix2Pix (c) HyperRIM (d) Target

Figure 8: Visualization of compressed input, decompressed output images from Pix2Pix and our method (HyperRIM) and the observed target image. As shown, the Pix2Pix output contains large pixel blocks whereas HyperRIM output successfully removes most artifacts.

6 Related Work

The proposed problem setting is related to multi-label prediction (Hsu et al., 2009) and mixture regression (Wedel & Kamakura, 2000). Both aim to predict multiple targets. In the former, the labels are usually discrete and multiple labels per input are given as supervision. In the latter, while the labels are continuous, a fixed number of modes is assumed for every input.

In terms of the underlying technique, the proposed approach relies on implicit generative models, and so related are work on GANs (Goodfellow et al., 2014; Gutmann et al., 2014; Mirza & Osindero, 2014; Odena et al., 2017; Isola et al., 2017) and IMLE (Li & Malik, 2018; Li et al., 2020).

In terms of the tasks, there is a large body of work on super-resolution, most of which consider upscaling factors of $2 - 4\times$. See (Yang et al., 2014; Nasrollahi & Moeslund, 2014; Wang et al., 2020) for comprehensive surveys. Most relevant are methods based on regression and conditional GANs, such as (Dong et al., 2014; Kim et al., 2016; Ledig et al., 2017; Sajjadi et al., 2017; Wang et al., 2018a). However, they can only produce a single output for the same input, either due to the deterministic nature of the model or mode collapse. Also related are methods that progressively upscale the input through a number of intermediate resolutions, e.g.: (Park et al., 2018; Lai et al., 2018; Wang et al., 2018b). Concurrently to this work, there has been work on extreme super-resolution which tries to upscale a fairly large image by $16\times$ (Shang et al., 2020), for which no implementation is publicly available. The challenges are however different, because the input image already contains rich structure and a fair amount of details.

There is relatively little work on image decompression to our knowledge; however, more work was done on image compression (Agustsson et al., 2018; 2019), which changes the encoding of the compressed image itself.

7 Conclusion

In this paper, we considered a setting where the prediction problem is inherently one-to-many, but where the supervision is only one-to-one. This differs from traditional settings like regression or class-conditional generative modelling – in the former, both prediction and supervision are one-to-one, whereas in the latter, both are one-to-many. We explored several problems with this characteristic and demonstrated that our approach was able to generate different plausible outputs for the same input, even though only one output per input is available as supervision. Moreover, we introduced an architecture for IMLE which outperformed GAN-based methods and can offer benefits like training stability and the lack of mode collapse.

References

Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Extreme learned image compression with gans. In *CVPR Workshops*, volume 1, pp. 2, 2018.

Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 221–231, 2019.

Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM Multimedia Systems Conference*, pp. 219–224, 2015.

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

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Michael U Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Likelihood-free inference via classification. *arXiv preprint arXiv:1407.4981*, 2014.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pp. 6626–6637, 2017.

Daniel J Hsu, Sham M Kakade, John Langford, and Tong Zhang. Multi-label prediction via compressed sensing. In *Advances in neural information processing systems*, pp. 772–780, 2009.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1646–1654, 2016.

Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2599–2613, 2018.

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, pp. 4, 2017.

Ke Li and Jitendra Malik. Fast k-nearest neighbour search via Dynamic Continuous Indexing. In *International Conference on Machine Learning*, pp. 671–679, 2016.

Ke Li and Jitendra Malik. Fast k-nearest neighbour search via Prioritized DCI. In *International Conference on Machine Learning*, pp. 2081–2090, 2017.

Ke Li and Jitendra Malik. Implicit maximum likelihood estimation. *arXiv preprint arXiv:1809.09087*, 2018.

Ke* Li, Shichong* Peng, Tianhao* Zhang, and Jitendra Malik. Multimodal image synthesis with conditional implicit maximum likelihood estimation. *International Journal of Computer Vision*, May 2020. ISSN 1573-1405. doi: 10.1007/s11263-020-01325-y. URL https://doi.org/10.1007/s11263-020-01325-y.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

Kamal Nasrollahi and Thomas B. Moeslund. Super-resolution: a comprehensive survey. *Machine Vision and Applications*, 25:1423–1468, 2014.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pp. 271–279, 2016.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pp. 2642–2651, 2017.

Dongwon Park, Kwanyoung Kim, and Se Young Chun. Efficient module based single image super resolution for multiple problems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 882–890, 2018.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Mehdi S. M. Sajjadi, Bernhard Schölkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4501–4510, 2017.

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *ArXiv*, abs/1602.07868, 2016.

Taizhang Shang, Qiuju Dai, Shengchen Zhu, Tong Yang, and Yandong Guo. Perceptual extreme super resolution network with receptive field block. *arXiv preprint arXiv:2005.12597*, 2020.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pp. 4790–4798, 2016.

Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. Esrgan: Enhanced super-resolution generative adversarial networks. *CoRR*, abs/1809.00219, 2018a.

Yifan Wang, Federico Perazzi, Brian McWilliams, Alexander Sorkine-Hornung, Olga Sorkine-Hornung, and Christopher Schroers. A fully progressive approach to single-image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 864–873, 2018b.

Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Michel Wedel and Wagner A Kamakura. Mixture regression models. In *Market segmentation*, pp. 101–124. Springer, 2000.

Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*, 2016.

Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In *Proceedings of European Conference on Computer Vision*, 2014.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *arXiv preprint*, 2018.

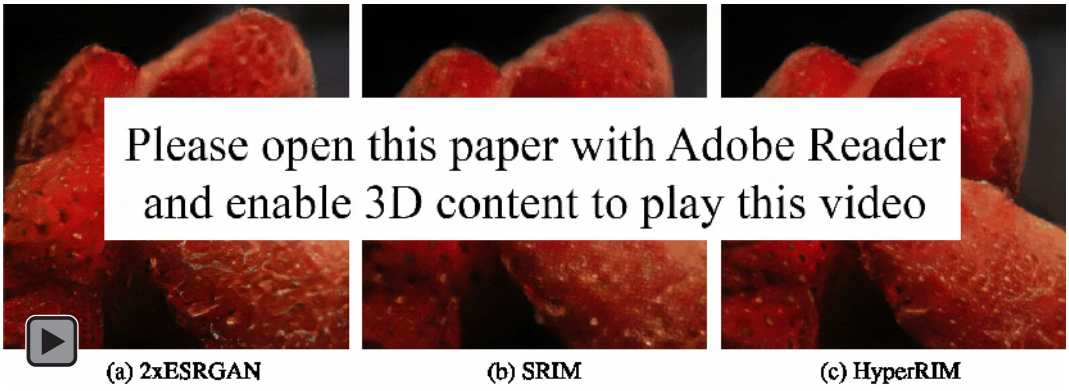


Figure 9: Visualization of output images from each method while traversing the space of random codes using gradient descent to reach the observed output image. As shown, (a) fails to reach the observed output, (b) comes close to the observed output but does not quite reach it, and (c) reaches the observed output and only encounters images with plausible content and texture. This reveals both the (i) precision and (ii) recall of each method, i.e.: the ability of each method to generate (i) *only* plausible images and (ii) *all* plausible images, which include the observed output. Since (a) cannot generate the observed output, its recall is low. Since (b) cannot reach the observed output, its recall is also unsatisfactory. Since (c) can reach the observed output and does so smoothly without generating an implausible image, the recall and precision of (c) are high.

A Precision and Recall

In Figure 9, we evaluate the precision and recall of each method, i.e.: whether the trained model can generate (a) *only* valid outputs, and (b) *all* valid outputs. Since only images that have a corresponding latent code $\mathbf{z}$ can be generated, we can explore the space of latent codes, which should be equivalent to the space of images that can be generated. We perform the following experiment: for a test image, we optimize over the latent code to try to find an image that is as close as possible to the original high-resolution image as measured by LPIPS and visualize the images we encounter along the way. For an ideal model, traversing the space of latent codes should (a) *only* pass through valid outputs (i.e. achieves high precision), and (b) be able to reach *any* valid image, including the original image (i.e.: achieves high recall). We find that HyperRIM is able to achieve better precision and recall than the baselines.

B Conditional IMLE Pseudocode

Algorithm 1 Conditional IMLE Training Procedure

Require: The set of inputs $\{\mathbf{x}_i\}_{i=1}^n$ and the set of corresponding observed outputs $\{\mathbf{y}_i\}_{i=1}^n$
  Initialize the parameters θ of the generator T_θ
  for $p = 1$ **to** N **do**
    Pick a random batch $S \subseteq \{1, \ldots, n\}$
    for $i \in S$ **do**
      Randomly generate i.i.d. m latent codes $\mathbf{z}_1, \ldots, \mathbf{z}_m$
      $\tilde{\mathbf{y}}_{i,j} \leftarrow T_\theta(\mathbf{x}_i, \mathbf{z}_j)\ \forall j \in [m]$
      $\sigma(i) \leftarrow \arg\min_j d(\mathbf{y}_i, \tilde{\mathbf{y}}_{i,j})\ \forall j \in [m]$
    end for
    for $q = 1$ **to** M **do**
      Pick a random mini-batch $\tilde{S} \subseteq S$
      $\theta \leftarrow \theta - \eta \nabla_\theta \left(\sum_{i \in \tilde{S}} d(\mathbf{y}_i, \tilde{\mathbf{y}}_{i,\sigma(i)}) \right) / |\widetilde{S}|$
    end for
  end for
  return θ

(a) Input

(b) Bicubic

(c) 2xESRGAN

(d) SRIM

(e) HyperRIM

(f) Observed Output

C MORE SAMPLES

(a) Input

(b) Bicubic

(c) 2xESRGAN

(d) SRIM

(e) HyperRIM

(f) Observed Output

(a) Input

(b) Bicubic

(c) 2xESRGAN

(d) SRIM

(e) HyperRIM

(f) Observed Output

(a) Input

(b) Bicubic

(c) 2xESRGAN

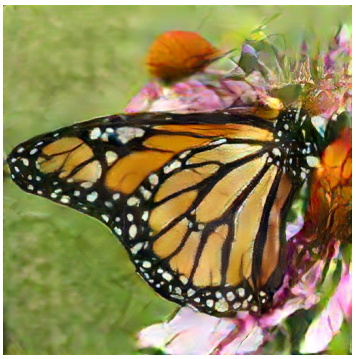
(d) SRIM

(e) HyperRIM

(f) Observed Output

(a) Input

(b) Bicubic



(c) 2xESRGAN

(d) SRIM

(e) HyperRIM

(f) Observed Output

(a) Input

(b) Bicubic

(c) 2xESRGAN

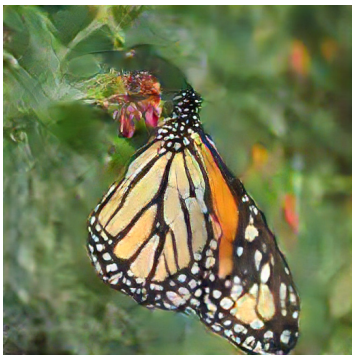
(d) SRIM

(e) HyperRIM

(f) Observed Output

(a) Input

(b) Bicubic



(c) 2xESRGAN

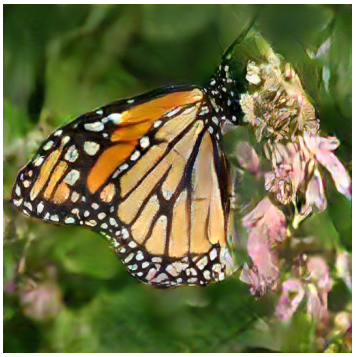
(d) SRIM

(e) HyperRIM

(f) Observed Output

(a) Input

(b) Bicubic



(c) 2xESRGAN

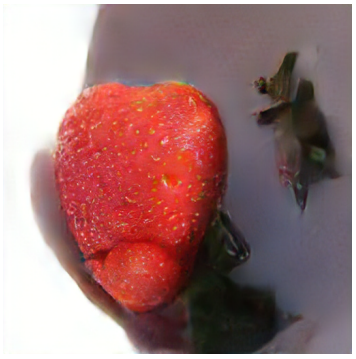
(d) SRIM

(e) HyperRIM

(f) Observed Output

(a) Input

(b) Bicubic



(c) 2xESRGAN

(d) SRIM

(e) HyperRIM

(f) Observed Output

(a) Input (b) Bicubic (c) 2xESRGAN

(d) SRIM (e) HyperRIM (f) Observed Output

(a) Input (b) Bicubic (c) 2xESRGAN

(d) SRIM (e) HyperRIM (f) Observed Output

(a) Input (b) Bicubic (c) 2xESRGAN

(d) SRIM (e) HyperRIM (f) Observed Output