GENERATING UNOBSERVED ALTERNATIVES WITH TOWER IMPLICIT MODEL (TIM)

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 and develop an approach, TIM, that can generate multiple high quality predictions given the same input and achieves a reduction of the Fréchet Inception Distance (FID) by 19.6% on average compared to the baseline.

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 to 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.

When do such prediction problems arise? They often come up in inverse problems, which require generating *more* information from *less* information, including information that cannot be derived from the input. 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.

Extending GAN-based approaches to the one-to-many prediction has proven to be challenging (Isola et al., 2017; Zhu et al., 2017) – due to mode collapse, the generator tends to generate identical samples for the same input and ignores the latent noise. A recent method (Li* et al., 2020) takes a different approach by extending an alternative generative modelling technique known as Implicit Maximum Likelihood Estimation (IMLE) (Li & Malik, 2018). While it shows promise in terms of generation diversity, it exhibits several major limitations: (1) the fidelity of generated images is

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.

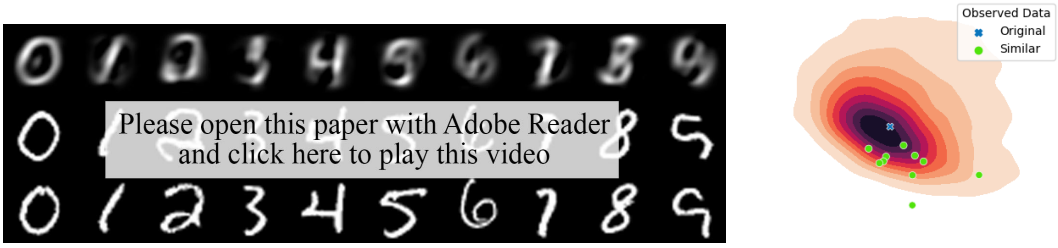
lacking, (2) it used a different dedicated architecture for each of the two tasks it considered, which limits its applicability to other tasks, (3) a large number of samples need to be drawn during training to attain high generation quality, which slows down training.

In this paper, we propose a new method called Tower Implicit Model (TIM) which produces different alternative predictions for the same input and addresses the aforementioned issues. Our contribution is three-fold:

1. We improve the quality of generated images significantly by leveraging cues at multiple scales and introducing intermediate supervision
2. We devise a single unified architecture that works well for a diverse range of tasks and is easily extensible to different output resolutions
3. We propose a new sampling scheme for IMLE which attains significantly greater efficiency

We demonstrate TIM significantly outperforms the prior method (Li* et al., 2020) in terms of both quality and diversity on a variety of tasks. Moreover, we show TIM achieves superior image quality compared to leading task-specific methods.

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 original images. Video is also available in supplementary materials.

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. The input is the image reconstructed from the first ten coordinates on the PCA basis 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 original images.

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.

3 BACKGROUND

In ordinary one-to-one prediction, the model is a function f_θ parameterized by θ that maps the input to the prediction. To support one-to-many prediction, one can add a latent random variable as an input, so now f_θ takes in both the input $\mathbf{x}$ and a latent noise vector $\mathbf{z}$ drawn from a standard Gaussian $\mathcal{N}(0, \mathbf{I})$ and produces an image $\widehat{\mathbf{y}}$ as output. In the language of generative models, f_θ is known as a *generator*. To train such a model, we can use a conditional GAN (cGAN), which adds a discriminator that tries to tell apart the observed output $\mathbf{y}$ and the generated output $\widehat{\mathbf{y}}$. The generator is trained to make its output $\widehat{\mathbf{y}}$ seem as real as possible to the discriminator. Unfortunately, after training, $f_\theta(\mathbf{x}, \mathbf{z})$ produces the same output for all values of $\mathbf{z}$ because of mode collapse, making conditional GANs ill-suited to the present problem setting. Intuitively, this happens because making $\widehat{\mathbf{y}}$ as real as possible would push it towards the observed output $\mathbf{y}$, so the generator tries to make its output similar to the observed output $\mathbf{y}$ for all values of $\mathbf{z}$.

In (Li* et al., 2020), an alternative technique is proposed to train the generator network f_θ, which is known as conditional IMLE (cIMLE). Rather than trying to make *all* outputs generated from different values of $\mathbf{z}$ similar to the observed output $\mathbf{y}$, it only tries to make *some* of them similar to the observed output $\mathbf{y}$. The generator is therefore only encouraged to map one value of $\mathbf{z}$ to the observed output $\mathbf{y}$, and reserve other values of $\mathbf{z}$ to *other* reasonable outputs that are not in the training dataset. This makes it possible to produce non-deterministic prediction. Also, unlike cGANs, cIMLE does not use a discriminator and therefore does not require adversarial training, which makes training more stable. The following training objective takes the following form:

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

where $d(\cdot,\cdot)$ is a distance metric, m is a hyperparameter, and $\mathbf{x}_i$ and $\mathbf{y}_i$ are the ith input and observed output in the dataset.

Unfortunately, in (Li* et al., 2020), a different generator network was used for each task, thereby limiting its applicability more generally to other tasks. Moreover, the generated outputs have low fidelity and lack fine details, especially when compared to one-to-one methods like cGANs, raising the question of whether cIMLE can produce images of comparable fidelity to cGANs despite having no discriminator. In this paper, we address these issues and answer the latter question in the affirmative.

4 METHOD

4.1 LEVERAGING MULTIPLE SCALES

Images contain structure at different scales, and it is important to both leverage cues at multiple scales in the input image and produce realistic global structure and fine details in the output image. To this end, we devise a meta-architecture suited to modelling structure at multiple scales. This high-level theme of multi-scale processing isn't new and has been used by many methods in various

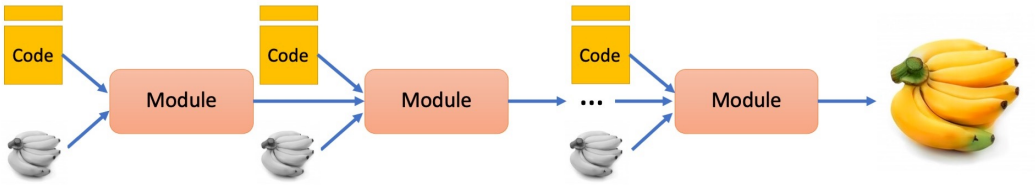


Figure 2: Our TIM model consists of multiple modules, each of which operates on $2\times$ the resolution of previous one.

contexts, e.g.: (Denton et al., 2015; Newell et al., 2016; Karras et al., 2017; Chen & Koltun, 2017; Park et al., 2018). What is interesting is the precise way this is done in order to enable intermediate supervision and hierarchical sampling, which are described below. It turns out both are critically important to achieving high fidelity generation, which is validated by an ablation study in Sect. 5.3.

In our architecture (shown in Figure 2), we have a sequence of modules, each of which handles an input image of a particular resolution and outputs an image of the same resolution. We downsample the input image repeatedly by a factor of 2 to obtain a set of input images at different resolutions and feed them into different modules. Each module takes a latent code whose spatial dimensions correspond to its resolution and the upsampled output of the module for the next lowest resolution as input. Note that this architecture generalizes to varying levels of resolution, since we can simply add more modules for high-resolution outputs.

We add supervision to the output of each intermediate module to encourage similarity between the generated image and the real image. Effectively, the distance metric in cIMLE is chosen to be the *sum* over perceptual distances between the output of each module and the real image downsampled to the same resolution. We choose LPIPS (Zhang et al., 2018) as our perceptual distance metric.

4.2 Hierarchical Sampling

Recall that for each input, cIMLE generates many samples and tries to make one of them similar to the observed output. The samples that are not selected correspond to the other possible outputs that are unobserved. So, the more samples that are generated during training, the more modes of the output distribution cIMLE can model. While we would ideally like to use many samples during training, generating samples is expensive, and so in practice, we can only generate just enough samples for cIMLE to learn effectively. This forces a tradeoff between the number of samples and performance, which is less than ideal.

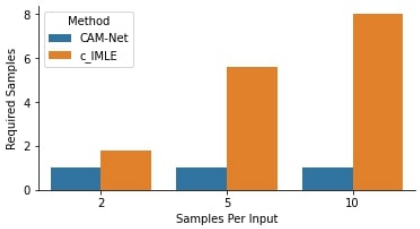


Figure 3: Comparison of sample efficiency of hierarchical sampling (HS) to vanilla sampling (which samples latent codes for different modules independently). The relative disparity of the required number of samples needed to achieve the same LPIPS distance to the observed output with/without HS is shown, where the number of required samples for HS is normalized to 1. The reported results are averaged over 10 independent runs. As shown, as the number of samples used per module increases in the case of HS, more samples are needed by vanilla sampling to match the distance attained by HS.

To get around this conundrum, we propose a novel sampling strategy, known as hierarchical sampling. Because cIMLE only uses the sample that is closest to the observed output for training, the key idea is to sample close to the region of the latent code space that is likely to be close to the observed output. This avoids generating samples that are unlikely to be selected, thereby increasing the effective number of samples without actually generating all of them.

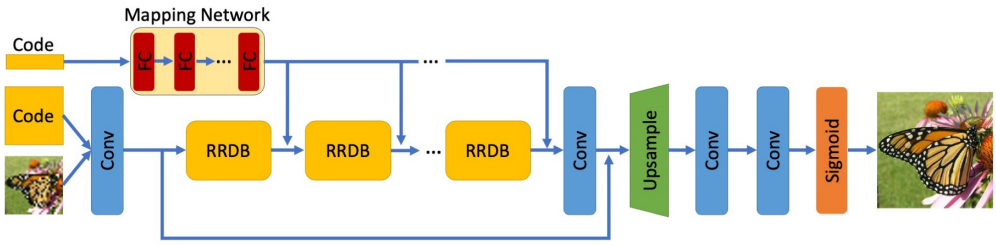
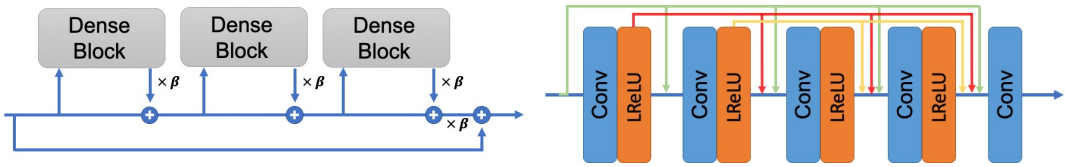


Figure 4: Details of the architecture backbone. See Figure 5a for the inner workings of RRDB blocks.



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

Figure 5: (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.

To this end, we generate samples for different modules successively, each of which operates at different resolutions. In the first stage, we sample latent codes for the first module, generate low-resolution images from them using the first module and select the latent code whose generated image is closest to the observed output downsampled to the appropriate resolution. In the next stage, we condition on the latent code of first module by setting it to the latent code selected in the previous stage. We sample latent codes for the second module and generate images at the next higher resolution from them using the first and second modules. In subsequent stages, we repeat the analogous procedure for the later modules. Note that this procedure is only used at train time; at test time, the latent codes for different modules are sampled independently because the goal at test time is to generate all possible outputs, including those that are unobserved.

We validate the improved sample efficiency of hierarchical sampling in Figure 3. We compare the number of samples required to obtain the same level of LPIPS distance to the observed output, with and without hierarchical sampling. As shown, vanilla sampling requires 2 to 8 times more samples than hierarchical sampling to reach the same LPIPS distance, and the difference becomes larger as the number of samples used for each module in hierarchical sampling increases.

4.3 Unified Model Architecture

In the generative modelling literature, architecture design has played an important role in advancing image fidelity (Radford et al., 2015; Reed et al., 2017; Vahdat & Kautz, 2020b), and different types of generative models have different optimal choices of architecture due to differences in the goal (mode seeking vs. mode covering) and training objective (adversarial vs. non-adversarial).

Prior cIMLE architectures are unable to generate fine details, so to generate high fidelity images, we design a new architecture for cIMLE. Unlike prior cIMLE architectures (Li* et al., 2020), the proposed architecture performs well across a broad variety of image synthesis tasks; in fact, the single proposed architecture significantly outperforms prior task-specific architectures, as shown later in Sect. 5.

In each module, the backbone architecture comprises of two branches, a main branch consisting of residual-in-residual dense blocks (RRDB) (Wang et al., 2018b) and an auxiliary branch consisting of a sequence of dense layers, known as a mapping network (Karras et al., 2019b), that produces scaling factors and offsets for different channels in output produced by each RRDB. In Figure 5, we show the inner workings of each RRDB, which is made up of dense blocks and residual connections. Unlike traditional RRDB, which comes without normalization and deliberately removes

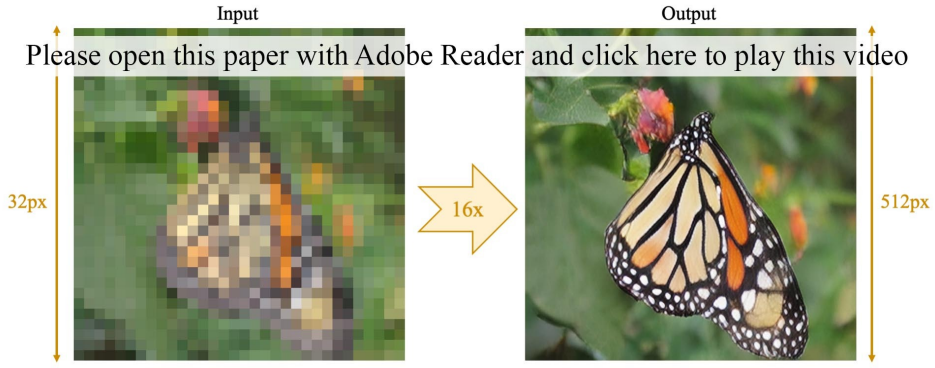


Figure 6: Visualization of different samples generated by our method (TIM) and the input for super-resolution. As shown, TIM generates different high quality textures for example on the edge of the butterfly's wing.

batch normalization in particular, we apply weight normalization (Salimans & Kingma, 2016) to all convolution layers.

Figure 7: Visualization of different samples generated by our method (TIM) and the input for image colourization. As shown in the figure, in addition to common colours, TIM also produces a variety of colours, such as green bananas and plums. Similarly, generating parrots with different body colours also shows the power of TIM in terms of multimodality.

The precise design of the architecture is essential to achieving high image fidelity across various tasks. For example, we found the default number of blocks and channels in (Wang et al., 2018b) to work poorly with IMLE, and needed to increase the number of channels and decrease the number of blocks at the same time. Arriving at the sweet spot in the space of architectures required thorough experimentation. As we will show in Sect. 5 and the appendix, the combination of the design motifs

(a) Input (b) TIM

Figure 8: Visualization of different samples generated by our method (TIM) and the input for image decompression. As shown, TIM output successfully removes most artifacts and predicts diverse textures.

leads to a substantial improvement in generated image fidelity compared to prior architectures. We also include an ablation study of different components to show the effectiveness of each in Sect. 5.3.

5 Experiments

We apply our method to four different one-to-many prediction tasks, namely $16\times$ single image super-resolution, image colourization, image decompression and image synthesis from scene layouts. We compare to the leading one-to-many prediction method based on IMLE, cIMLE (Li* et al., 2020), which serves to validate our main contributions, namely improving the fidelity, generality and efficiency of IMLE-based methods. As a secondary comparison, we also compare to the leading task-specific one-to-many method for each task, to demonstrate potential impact in a broader context. Where a task-specific one-to-many method does not exist for a task, we compare instead to the leading one-to-one method. Where there are a large number of methods that fit the criterion, we picked the leading method with publicly available implementation based on leaderboard rankings in challenges and recent survey papers (Zhang et al., 2020; Anwar et al., 2020).

5.1 Quantitative Results

In one-to-many prediction, diversity of predictions is most important, since without diversity, one-to-many prediction becomes the same as one-to-one prediction. We evaluate output diversity using faithfulness-weighted variance (Li* et al., 2020) and LPIPS diversity score (Zhu et al., 2017). LPIPS diversity score is the average LPIPS distance between different output samples for the same input, whereas faithfulness-weighted variance is the LPIPS distance between the output samples and the mean, weighted by the consistency with the target output measured by a Gaussian kernel. The kernel bandwidth parameter σ trades off the importance of consistency vs. diversity.

We evaluate perceptual quality of predictions using the Fréchet Inception Distance (FID) (Heusel et al., 2017), since classical metrics like PSNR and SSIM do not capture perceptual quality well (Ledig et al., 2017).

We compare the perceptual quality and output diversity in Tables 2, 3 and 4. As shown in Table 2, TIM outperforms both the one-to-many prediction baseline, cIMLE, and specialized one-to-one prediction baselines, in terms of FID. As shown in Tables 3 and 4, TIM outperforms the one-to-many prediction baseline, cIMLE, in terms of faithfulness-weighted variance at all bandwidth parameters and LPIPS diversity score for all tasks. Comparisons to one-to-one prediction baselines in terms of faithfulness-weighted variance are not shown explicitly because their faithfulness-weighted variances are zero. These comparisons indicate that TIM is able to produce more realistic and diverse images than the baselines.

5.2 Qualitative Results

We show the results of our method and the input for super-resolution in Figure 6, colourization in Figure 7 and image decompression in Figure 8. Results for image synthesis from scene layouts and

	Super-Resolution			Image Decompression		
	TIM	*cIMLE*	*RFB-ESRGAN*	*TIM*	*cIMLE*	*DnCNN*
FID	**16.75**	27.34	19.56	**72.75**	100.48	109.38

	Colourization				
	TIM	*cIMLE*	*Zhang et al.*	*Iizuka et al.*	*Larsson et al.*
FID	**33.19**	36.38	57.95	85.88	47.44

Table 2: Comparison of fidelity of generated images, measured by the Fréchet Inception Distance (FID) between the observed images and the samples generated by our method (TIM) and the leading IMLE-based and task-specific baselines. Lower values of FID are better. We compare favourably relative to the baselines.

	Super-Resolution		Image Decompression		Colourization		
σ	*TIM*	*cIMLE*	*TIM*	*cIMLE*	*TIM*	*cIMLE*	*Zhang et al.*
0.3	**.0572**	.0548	**.0513**	.0493	**.124**	.105	.0789
0.2	**.00586**	.00522	**.00380**	.00314	**.0621**	.0456	.0318
0.15	**.000344**	.000273	**.000223**	.000132	**.0284**	.0179	.0110

Table 3: Comparison of faithfulness weighted variance of the samples generated by our method (TIM) and other one-to-many baselines on different tasks. Higher value shows more variation in the generated samples that are faithful to the original image. σ is the bandwidth parameter for the Gaussian kernel used to compute the faithfulness weights.

comparisons to the baselines are included in the appendix. As shown, TIM generates high quality and diverse results.

5.3 Ablation Study

We incrementally remove (1) hierarchical sampling (HS), (2) mapping network (MN), (3) intermediate supervision (IS), (4) weight normalization (WN). As shown in Figure 9 and Table 5, each component is critical to achieving best results.

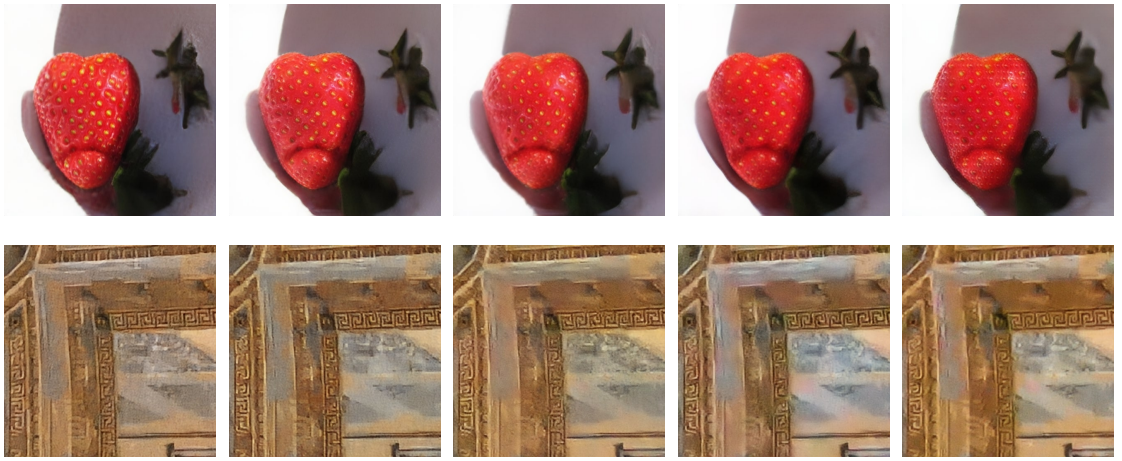
6 Related Work

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 autoregressive models (Salimans et al., 2017; van den Oord et al., 2016b;a), VAEs (Kingma & Welling, 2014; Vahdat & Kautz, 2020a; Child, 2020; Razavi et al., 2019), GANs (Goodfellow et al., 2014; Karras et al., 2019a; Brock et al., 2019; Karras et al., 2020), normalizing flows (Dinh et al., 2017; Kobyzev et al., 2020; Kingma & Dhariwal, 2018), energy based methods (Ackley et al., 1985; Du & Mordatch, 2019; Nijkamp et al., 2019; Zhao et al., 2021; Xie et al., 2021a;b), score-based models (Song & Ermon, 2019; Ho et al., 2020; Jolicoeur-Martineau et al., 2021) and IMLE (Li & Malik, 2018; Li* et al., 2020).

	Super-Resolution		Image Decompression		Colourization		
	TIM	*cIMLE*	*TIM*	*cIMLE*	*TIM*	*cIMLE*	*Zhang et al.*
LPIPS Score	**.180**	.168	**.276**	.236	**.0364**	.0334	.0108

Table 4: Comparison of LPIPS diversity score of the samples generated by our method (TIM) and other one-to-many baselines on different tasks. Higher value shows more variation in the generated samples.



(f) Full Model (g) No HS (h) No HS, MN (i) No HS, MN, IS (j) No HS, MN, IS, WN

Figure 9: Visualization comparison of two tasks: $16\times$ super-resolution (first row) and image decompression (second row) as we gradually remove (1) hierarchical sampling (HS), (2) mapping network (MN), (3) intermediate supervision (IS), (4) weight normalization (WN).

	Full Model	No HS	No HS, MN	No HS, MN, IS	No HS, MN, IS, WN
SR	**16.75**	17.70	18.58	21.66	22.51
DC	**72.75**	81.24	84.57	99.54	102.82

Table 5: Comparison of Fréchet Inception Distance (FID) of two tasks: super-resolution (SR) and image decompression (DC) by gradually remove (1) hierarchical sampling (HS), (2) mapping network (MN), (3) intermediate supervision (IS), (4) weight normalization (WN).

There is a large body of work on task-specific methods. For super-resolution (Yang et al., 2014; Nasrollahi & Moeslund, 2014; Wang et al., 2020), most consider upscaling factors of $2-4\times$ and are based on direct regression (e.g.: (Dong et al., 2014)) or conditional GANs (e.g.: (Ledig et al., 2017)). Hence, they are one-to-one prediction methods. For colourization (Anwar et al., 2020), many are one-to-one methods and differ mostly in the architecture (e.g.: (Iizuka et al., 2016; Larsson et al., 2016)). A notable exception is (Zhang et al., 2016), which discretizes the colour space and learns a marginal distribution over the per-pixel colours. Image decompression is often treated as a denoising problem (Tian et al., 2018); most methods are based on direct regression and are one-to-one. They differ mostly in architecture (e.g.: (He et al., 2016; Zhang et al., 2017)). For image synthesis from scene layouts, most methods are one-to-one and GAN-based (e.g.: (Sun & Wu, 2019; Wang et al., 2018a)). Notable exceptions include (Chen & Koltun, 2017) (mixture of regression-based) and (Li* et al., 2020) (IMLE-based).

7 Conclusion

In this paper, we considered a setting where prediction is inherently one-to-many, but where 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 developed an improved method for this challenging problem based on the conditional IMLE (cIMLE) framework and addressed three main issues of the prior cIMLE-based approach: image fidelity, task-specific architectures and sample efficiency. We proposed a new method, TIM, which is a single architecture that can be applied to a broad variety of tasks. The modularity of the architecture allows us to devise a novel hierarchical sampling scheme for IMLE, which improves the sample efficiency. Finally, we demonstrate our single architecture can generate significantly higher fidelity output images without compromising on diversity.

8 Ethics Statement

Because TIM can be applied to a broad range of tasks, its societal impact depends on the applications that it is used for. While most applications are harmless, it could be potentially used for tasks like demosaicing which have privacy or copyright implications.

9 Reproducibility Statement

We include our source code in the supplementary material.

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985. 8

Saeed Anwar, Muhammad Tahir, Chongyi Li, A. Mian, F. Khan, and A. W. Muzaffar. Image colorization: A survey and dataset. *ArXiv*, abs/2008.10774, 2020. 7, 9

S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, 2002. doi: 10.1109/TPAMI.2002.1033210. 14

Andrew Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *ArXiv*, abs/1809.11096, 2019. 8

Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, pp. 3, 2017. 4, 9

R. Child. Very deep vaes generalize autoregressive models and can outperform them on images. *ArXiv*, abs/2011.10650, 2020. 8

Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pp. 1486–1494, 2015. 4

Laurent Dinh, Jascha Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. *ArXiv*, abs/1605.08803, 2017. 8

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. 9

Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019. 8

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. 8

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. 9

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. 7

Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020. 8

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. 8

S. Iizuka, Edgar Simo-Serra, and H. Ishikawa. Let there be color! *ACM Transactions on Graphics (TOG)*, 35:1 – 11, 2016. 9

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. 1

Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021. 8

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 4

Tero Karras, S. Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4396–4405, 2019a. 8

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, 2019b. 5

Tero Karras, S. Laine, Miika Aittala, Janne Hellsten, J. Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8107–8116, 2020. 8

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018. 8

Diederik P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014. 8

I. Kobyzev, Simon Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 8

Gustav Larsson, M. Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. *ArXiv*, abs/1603.06668, 2016. 9

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. 7, 9

Ke Li and Jitendra Malik. Implicit maximum likelihood estimation. *arXiv preprint arXiv:1809.09087*, 2018. 1, 8

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. 1, 2, 3, 5, 7, 8, 9, 22

Kamal Nasrollahi and Thomas B. Moeslund. Super-resolution: a comprehensive survey. *Machine Vision and Applications*, 25:1423–1468, 2014. 9

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 4

Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model. *arXiv preprint arXiv:1904.09770*, 2019. 8

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. 4

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 5

Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *NeurIPS*, 2019. 8

Scott Reed, Aäron Oord, Nal Kalchbrenner, Sergio Gómez Colmenarejo, Ziyu Wang, Yutian Chen, Dan Belov, and Nando Freitas. Parallel multiscale autoregressive density estimation. In *International Conference on Machine Learning*, pp. 2912–2921. PMLR, 2017. 5

Stephan R. Richter, Vibhav Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. *ArXiv*, abs/1608.02192, 2016. 22

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *ArXiv*, abs/1602.07868, 2016. 6

Tim Salimans, A. Karpathy, Xi Chen, and Diederik P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *ArXiv*, abs/1701.05517, 2017. 8

Yang Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *ArXiv*, abs/1907.05600, 2019. 8

Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10530–10539, 2019. 9

Chunwei Tian, Yong Xu, Lunke Fei, and Ke Yan. Deep learning for image denoising: A survey. *ArXiv*, abs/1810.05052, 2018. 9

Arash Vahdat and J. Kautz. Nvae: A deep hierarchical variational autoencoder. *ArXiv*, abs/2007.03898, 2020a. 8

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020b. 5

Aäron van den Oord, Nal Kalchbrenner, Lasse Espeholt, K. Kavukcuoglu, Oriol Vinyals, and A. Graves. Conditional image generation with pixelcnn decoders. In *NIPS*, 2016a. 8

Aäron van den Oord, Nal Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *ArXiv*, abs/1601.06759, 2016b. 8

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8798–8807, 2018a. 9

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, 2018b. 5, 6

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. 9

Michel Wedel and Wagner A Kamakura. Mixture regression models. In *Market segmentation*, pp. 101–124. Springer, 2000. 8

Jianwen Xie, Zilong Zheng, Xiaolin Fang, Song-Chun Zhu, and Ying Nian Wu. Cooperative training of fast thinking initializer and slow thinking solver for conditional learning. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2021a. 8

Jianwen Xie, Zilong Zheng, Xiaolin Fang, Song-Chun Zhu, and Ying Nian Wu. Learning cycle-consistent cooperative networks via alternating mcmc teaching for unsupervised cross-domain translation. In *AAAI*, 2021b. 8

Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In *Proceedings of European Conference on Computer Vision*, 2014. 9

K. Zhang, W. Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26:3142–3155, 2017. 9

K. Zhang, Shuhang Gu, Radu Timofte, Taizhang Shang, Qiuju Dai, Shengchen Zhu, Tong Yang, Yandong Guo, Younghyun Jo, Sejong Yang, Seon Joo Kim, Lin Zha, Jiande Jiang, Xinbo Gao, Wen Lu, Jing Liu, Kwangjin Yoon, Taegyun Jeon, Kazutoshi Akita, Takeru Ooba, Norimichi Ukita, Zhipeng Luo, Yuehan Yao, Z. Xu, Dongliang He, Wenhao Wu, Yukang Ding, Chao Li, Fu Li, Shilei Wen, Jianwei Li, Fuzhi Yang, Huan Yang, Jianlong Fu, Byung-Hoon Kim, JaeHyun Baek, J. C. Ye, Yuchen Fan, Thomas S. Huang, Junyeop Lee, Bokyeung Lee, Jungki Min, Gwantae Kim, Kanghyu Lee, Jaihyun Park, Mykola Mykhailych, Haoyu Zhong, Yukai Shi, Xiaoju Yang, Zhijing Yang, Liang Lin, Tongtong Zhao, Jinjia Peng, Huibing Wang, Zhi Jin, Jiahao Wu, Yifu Chen, Chenming Shang, Huanrong Zhang, Jeongki Min, S HrishikeshP., Densen Puthussery, and V JijiC. Ntire 2020 challenge on perceptual extreme super-resolution: Methods and results. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2045–2057, 2020. 7

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, pp. 649–666. Springer, 2016. 9

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018. 4

Yang Zhao, Jianwen Xie, and Ping Li. Learning energy-based generative models via coarse-to-fine expanding and sampling. In *ICLR*, 2021. 8

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, pp. 465–476, 2017. 1, 7