CHIMLE: Conditional Hierarchical IMLE

Anonymous Author(s)
Affiliation
Address
email

Abstract

A persistent challenge in conditional image synthesis has been generating diverse output images from the same input image due to the problem of mode collapse. Implicit Maximum Likelihood Estimation (IMLE) is a recently proposed alternative that aims to address this issue. IMLE uses the same generator as GANs but adopts a different objective function which ensures each observed image has a generated sample nearby. To generate high-fidelity images, prior IMLE-based methods require a large number of samples. Doing so is expensive, and so this limits image fidelity in practice. In this paper, we propose a new method to get around this limitation, which we dub Conditional Hierarchical IMLE (CHIMLE), which can generate high-fidelity images without requiring many samples. We show on multiple tasks that CHIMLE significantly improves generated image fidelity, as demonstrated by a reduction in Fréchet Inception Distance (FID) by 36.9% on average compared to the prior best IMLE-based method.

1 Introduction

With the advent of deep learning, image synthesis has witnessed incredible progress in recent years. Advances in deep generative models such as GANs [27, 38, 42, 8, 43], VAEs [90, 14, 74, 48], autoregressive models [77, 92, 91], normalizing flows [20, 49, 47], energy-based models [1, 23, 67], score-based models [36, 82, 32, 40] have driven remarkable improvement in the fidelity of generated images with each successive generation, and it is now possible to synthesize novel and photorealistic images with these approaches.

Conditional image synthesis problems are often encountered in practice, where the goal is to generate images given some input that specifies what kinds of images should be generated. The conditioning input can be either discrete labels, like an object category label, or continuous labels, like another image. Among the latter class of problems, unimodal image synthesis is traditionally considered, which aims to generate an output image given an input image. A more challenging problem is *multimodal* conditional image synthesis, where the goal is to generate *multiple* plausible output images for the same given input image.

As a concrete example, suppose we are given a grayscale image of an apple and would like to generate colour versions of the image. The apple could be red or green, and the input grayscale image does not determine whether the apple should be red or green. Whereas the goal of unimodal synthesis is to generate either a red or a green apple, the goal of *multimodal* synthesis is to generate both a red apple and a green apple.

In this paper, we focus on one-shot generative models due to their fast inference speed as opposed to models that generate outputs iteratively, such as autoregressive models and score-based/diffusion models. The two most common families of one-shot generative models are VAEs and GANs. However, extending these two families of methods to the multimodal setting has proven to be challenging. VAEs suffer from posterior collapse [17, 60] where the encoder ignores the input images and causes

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

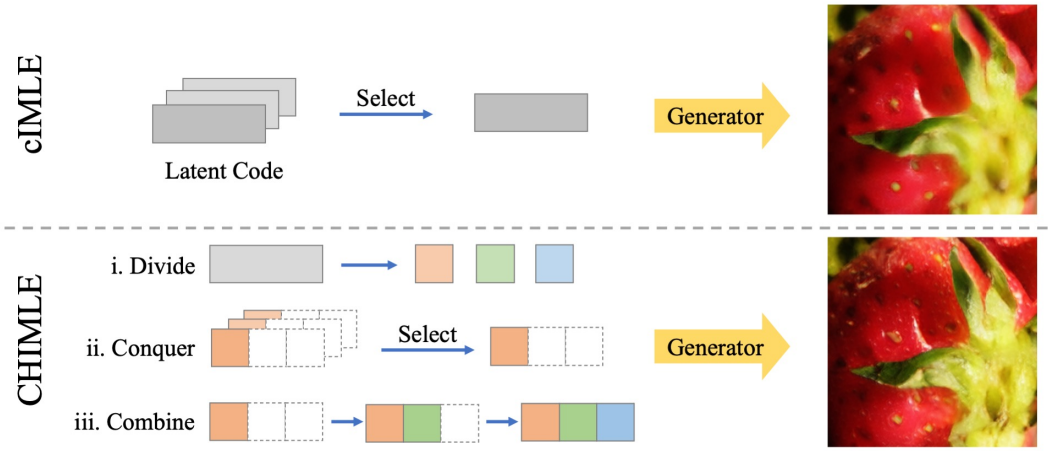


Figure 1: Overview of the prior best IMLE-based method, cIMLE [56] (top row) and the proposed method, CHIMLE (bottom row). Whereas cIMLE samples many latent codes and selects the one that produces the best result, CHIMLE divides the latent code by dimension into components, selects the best sample for each component and iteratively combines the selected components to form the overall latent code. As shown, the sample from CHIMLE contains more fine-grained details than the one from cIMLE.

the decoder to produce the conditional mean, which tends to be blurry. GANs suffer from mode collapse [38, 113] where the generator tends to ignore the latent noise and generate identical samples for the same input image. A recent method [56] takes a different approach by extending an alternative generative modelling technique known as Implicit Maximum Likelihood Estimation (IMLE) [55] to the conditional synthesis setting. At a high level, IMLE uses a generator like GANs, but instead of using an adversarial objective which makes each generated sample similar to some *observed image*, IMLE uses a different objective that ensures each observed image has some similar *generated samples*. Hence, the generator can cover the whole data distribution without dropping modes. We refer interested readers to [56, 55] for more details on the algorithm.

However, as shown in the top row of Figure 1, the output image from cIMLE lacks fine details. For cIMLE to generate high quality samples, there must a sample out of m samples generated during training that is similar to the observed image. This necessitates a large m; unfortunately, generating a large number of samples is computationally expensive. So we are faced with a tradeoff: do improvements in sample quality have to come at the expense of sample efficiency?

In this paper, we show that there is a way around this conundrum. The idea is to generate samples in a clever way such that the best sample is about as similar to the observed image as if a large number of samples had been generated, *without* actually generating that many samples. We propose three ideas: partitioning of the latent code, partial evaluation of latent code components and iterative construction of latent code as shown in the bottom row of Figure 1. These three ideas give rise to a novel method known as Conditional Hierarchical IMLE (CHIMLE). We demonstrate that CHIMLE significantly outperforms the prior best IMLE-based method [56] in terms of both fidelity and diversity on a variety of tasks. Moreover, we show that CHIMLE achieves superior image fidelity and mode coverage compared to leading general-purpose multimodal and task-specific methods.

2 Method

2.1 Preliminaries: Conditional Implicit Maximum Likelihood Estimation (cIMLE)

In ordinary unimodal synthesis, the model is a function f_θ parameterized by θ that maps the input to the generated output. To support multimodal synthesis, one can add a latent random variable as an input, so now f_θ takes in both the input $\mathbf{x}$ and a latent code $\mathbf{z}$ drawn from a standard Gaussian $\mathcal{N}(0, \mathbf{I})$ and produces an image $\widehat{\mathbf{y}}$ as output. To train such a network, we can use a conditional GAN (cGAN), which adds a discriminator network that tries to tell apart the observed image $\mathbf{y}$ 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, $f_\theta(\mathbf{x}, \mathbf{z})$ often produces the same output for all values of $\mathbf{z}$ because of mode collapse. Intuitively, this happens because making $\widehat{\mathbf{y}}$ as real as possible would push it towards the observed image $\mathbf{y}$, so the generator tries to make its output similar to the observed image $\mathbf{y}$ for all values of $\mathbf{z}$. As a result, naïvely applying cGANs to the problem of *multimodal* synthesis is difficult.

Conditional IMLE (cIMLE) [56] proposes an alternative technique for training the generator network f_θ. Rather than trying to make *all* outputs generated from different values of $\mathbf{z}$ similar to the observed image $\mathbf{y}$, it only tries to make *some* of them similar to the observed image $\mathbf{y}$. The generator is therefore only encouraged to map one value of $\mathbf{z}$ to the observed image $\mathbf{y}$, allowing other values of $\mathbf{z}$ map to *other* reasonable outputs that are not observed. This makes it possible to perform *multimodal* synthesis. Also, unlike cGANs, cIMLE does not use a discriminator and therefore obviates adversarial training, making training more stable. The cIMLE training objective is:

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

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 image in the dataset. During each iteration of the training loop, a *pool* of m samples are generated from f_θ for each input $\mathbf{x}^{(i)}$. The cIMLE algorithm then selects the *closest* sample for each observed image $\mathbf{y}^{(i)}$ to optimizes θ.

2.2 Conditional Hierarchical IMLE (CHIMLE)

In order to generate high-quality images using cIMLE, we need some of the m generated samples to be similar to the observed image $\mathbf{y}$ during training. Achieving this requires a large m, but generating samples is expensive and [56] can only generate just enough samples for cIMLE to learn effectively. This forces a tradeoff between the number of samples and the mode modelling accuracy, which is less than ideal.

The key observation is that the gradients of the model parameters depend solely on the *closest* sample for each observed image, as shown in Eqn. 1 – all other samples that are not selected do not contribute. Therefore, if we could efficiently search for a latent code $\mathbf{z}$ that would produce a sample of a comparable degree of similarity to $\mathbf{y}$ as one that would be selected from a larger pool $\mathbf{z}^*$, we could achieve high sample quality without actually generating that many samples.

To solve search problems, a time-tested paradigm in computer science is divide-and-conquer, which entails *dividing* the problem into simpler sub-problems, *conquering* each sub-problem and *combining* the solutions from the solved sub-problem. In the context of searching for a latent code $\mathbf{z}$ that simulates the quality of $\mathbf{z}^*$, we propose dividing the latent code into components, evaluating each latent code component and combining the latent code components iteratively.

In the ensuing discussion, we will consider a common paradigm of image synthesis models [73, 38, 12, 42], where the model processes the image at L different scales. We can feed different dimensions of the latent code as inputs to different layers operating at different resolutions, thereby controlling the variations of content at different resolutions.

Division of Latent Code To reduce the difficulty of the original problem, we can divide it into sub-problems. Each sub-problem should be a simpler instance of the original problem. The solution to one sub-problem should reveal some information on which other sub-problems to solve. In our specific context, one naïve way to divide the problem into sub-problems is to divide the latent space into regions. Each sub-problem would then involve searching for a $\mathbf{z}$ in a different region of the latent space that results in an output similar to the observed image. However, different regions of the latent space encode different appearances of the output (e.g., the particular output colour for a grayscale image of an apple as input), and solving a sub-problem can only reveal whether a particular region likely contains $\mathbf{z}^*$ (e.g., blue is not a plausible output colour), and so the solution to a sub-problem does not reveal clues on which other sub-problems to solve (e.g., the implausibility of blue does not inform whether red or green is plausible). Therefore, we propose to divide the dimensions of the latent code $\mathbf{z} \in \mathbb{R}^n$ into L different components $S = \{\mathbf{z}_i\}_{i=1}^L$, where $\mathbf{z}_i \in \mathbb{R}^{k_i}$ and $\sum_{i=1}^L k_i = n$, which amounts to dividing the latent space into mutually orthogonal subspaces. Each component $\mathbf{z}_i$ corresponds to the dimensions that serve as input to layers at the same resolution. By solving for the

appearances of lower resolutions (e.g., the coarse texture of an apple is red), we get clues about the probable appearances at higher resolutions (e.g., the fine details of the apple are likely red as well).

Partial Evaluation of Latent Code Components To determine how promising a latent code component is, we need a way to evaluate a partially constructed latent code, with only some components, indexed by $S_r \subset S$, being realized with concrete values. One way to achieve this is to set the remaining latent code components, indexed by $S_u = S \setminus S_r$, to zeros (i.e., $\mathbf{z}_j = \vec{\mathbf{0}}, \forall \mathbf{z}_j \in S_u$) or random values and evaluate the model output $\tilde{\mathbf{y}} = f_\theta(\mathbf{x}, \mathbf{z})$ against the observed image $\mathbf{y}$ by computing $d(\tilde{\mathbf{y}}, \mathbf{y})$. However, this is not ideal since the details generated from $\mathbf{z}_{S_u}$ in $\tilde{\mathbf{y}}$ are arbitrary, and so are likely to be very different from the corresponding details in $\mathbf{y}$. As a result, we may penalize a promising value of the realized components $\mathbf{z}_{S_r}$ by mistake.

We propose to use a partial output $\tilde{\mathbf{y}}_p$ that depends only on the currently constructed components $\mathbf{z}_{S_r}$, and evaluate $\tilde{\mathbf{y}}_p$ against a part of the observed image $\mathbf{y}_p$ that only contains details at resolutions modelled by $\mathbf{z}_{S_r}$. So instead of computing $d(\tilde{\mathbf{y}}, \mathbf{y})$, we compute $d(\tilde{\mathbf{y}}_p, \mathbf{y}_p)$. By doing so, we can evaluate the quality of $\mathbf{z}_{S_r}$ more accurately, since $\tilde{\mathbf{y}}_p$ does not depend on $\mathbf{z}_{S_u}$. To realize this, we add output heads at the L different resolutions that the model operates on to produce the partial outputs $\tilde{\mathbf{y}}_p$ and downsample the observed image $\mathbf{y}$ to the corresponding resolutions to produce the partial observed images $\mathbf{y}_p$. The values of the partially realized components that result in the greatest similarity to $\mathbf{y}_p$ is the final selected component, i.e., $\mathbf{z}_p^* = \arg\min d(\tilde{\mathbf{y}}_p, \mathbf{y}_p)$.

Algorithm 1 Conditional Hierarchical IMLE Algorithm for a Single Data Point

Require: One input image at L increasing resolutions $\{\mathbf{x}_l\}_{l=1}^L$, the set of partial observed images $\{\mathbf{y}_l\}_{l=1}^L$ at corresponding resolutions and the generator $f_\theta(\cdot, \cdot)$ that produces L outputs at corresponding resolutions

for $l = 1$ **to** L **do**

Randomly generate m i.i.d. latent codes for the lth component $\mathbf{z}_l^{(1)}, \ldots, \mathbf{z}_l^{(m)}$

$\tilde{\mathbf{y}}_l^{(j)} \leftarrow l^{\text{th}}$ output from $f_\theta((\mathbf{x}_1, \ldots, \mathbf{x}_l), (\mathbf{z}_1^*, \ldots, \mathbf{z}_{l-1}^*, \mathbf{z}_l^{(j)}))\ \forall j \in \{1, \ldots, m\}$

$\mathbf{z}_l^* \leftarrow \arg\min_j d(\mathbf{y}_l, \tilde{\mathbf{y}}_l^{(j)})\ \forall j \in \{1, \ldots, m\}$

end for

return $\{\mathbf{z}_l^*\}_{l=1}^L$

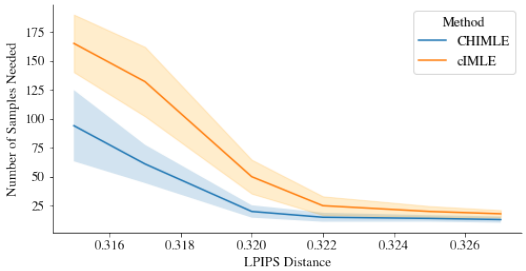


Figure 2: Generated sample quality comparison between the proposed Conditional Hierarchical IMLE (CHIMLE) and cIMLE [56]. The plot shows the number of generated samples needed to reach the same distance (as measured by LPIPS) to the observed image with CHIMLE or cIMLE. A smaller number of generated samples means better efficiency in finding a sample that is close to the observed image. The result is averaged over ten independent runs. As shown, CHIMLE consistently outperforms cIMLE by requiring significantly fewer samples. Also, the more stringent the required LPIPS distance gets, the greater the disparity between the number of samples needed by cIMLE vs. CHIMLE.

Iterative Combination of Latent Code Components To combine the optimal solution to each sub-problem and form a solution to the original problem, one strategy is to consider the sub-problems independently and aggregate the solution to each at the end. However, solutions to different sub-problems might be related, and so solving each independently would result in redundant work. It would be more efficient to avoid solving sub-problems with similar solutions, and instead focus on solving sub-problems that yield the greatest information gain given previously solved sub-problems. The challenge here is deciding which sub-problems to solve first and how to use their solutions to decide on which sub-problem to solve next. Since the amount of information in the image strictly increases as scale increases, solving the sub-problem at a higher resolution before the lower resolution would make the latter redundant. Therefore, we propose to start the construction of the latent code from the component operating at the lowest resolution to the highest, i.e. from $\mathbf{z}_1$ to $\mathbf{z}_L$, where $\mathbf{z}_i$'s are assumed to be ordered by resolution in increasing order. At a given resolution $l \in \{1 \ldots L\}$,

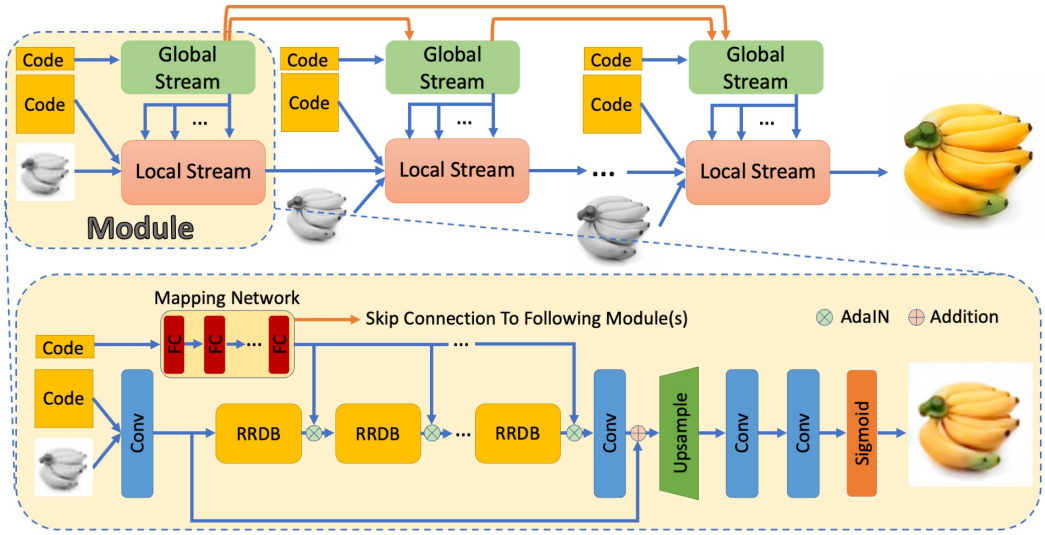


Figure 3: Our model consists of multiple modules, each of which operates on $2\times$ the resolution of the previous one. Each module contains a global stream and a local stream. The local stream consists of a sequence of residual-in-residual dense blocks (RRDB). The global stream consists of a mapping network which applies adaptive instance normalization (AdaIN) to the output of each RRDB block.

we set the components at all lower resolutions to the values selected in previous steps $\mathbf{z}_1^*, \ldots, \mathbf{z}_{l-1}^*$ to construct $\mathbf{z}_{S_l} = (\mathbf{z}_1^*, \ldots, \mathbf{z}_{l-1}^*, \mathbf{z}_l)$, and evaluate the quality of $\mathbf{z}_{S_l}$ using the partial evaluation method proposed above.

If we put everything together, we obtain the Conditional Hierarchical IMLE (CHIMLE) algorithm, which is detailed in Alg. 1 for the special case of one input image. We empirically validate the effectiveness of CHIMLE by comparing the number of generated samples CHIMLE needs to obtain the same level of similarity to the observed image as cIMLE. As shown in Figure 2, cIMLE consistently requires more samples than CHIMLE, demonstrating a significant improvement in sample efficiency.

2.3 Model Architecture

As shown in Figure 3, our architecture contains a sequence of modules, each of which handles an input image of a particular resolution and outputs an image whose resolution is increased by a factor of 2. 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. As introduced in Sect. 2.2, we divide the dimensions of the latent code into components and feed them to each of the modules. Note that this architecture generalizes to varying levels of output resolution, since we can simply add more modules for high-resolution outputs. We add intermediate output heads to each module and add supervision to the output of each module. We choose the LPIPS [109] distance as the distance metric used in the IMLE objective function.

For the design of each module, we follow prior works on modelling global and local image features [62, 9, 93, 61, 79] and design a module that comprises of two branches, a local processing branch and a global processing branch. The local processing branch takes in the conditional image input, a latent code with the matching resolution as the image and the output from the previous module if applicable. The global processing branch takes in a fixed-sized latent code input and produces scaling factors and offsets for the different channels of the intermediate features in the local processing branch. For more implementation details, please refer to the supplementary materials.

3 Experiments

Baselines To validate our main contribution of this paper, namely improving the output fidelity of IMLE-based methods, we compare to the leading IMLE-based method, cIMLE [56]. As a secondary comparison, we demonstrate potential impact in a broader context and compare our method to the leading multimodal general-purpose conditional image synthesis methods according to a recent survey [69]: BicyleGAN [113], MSGAN [63], DivCo [59] and MoNCE [26]. As a tertiary

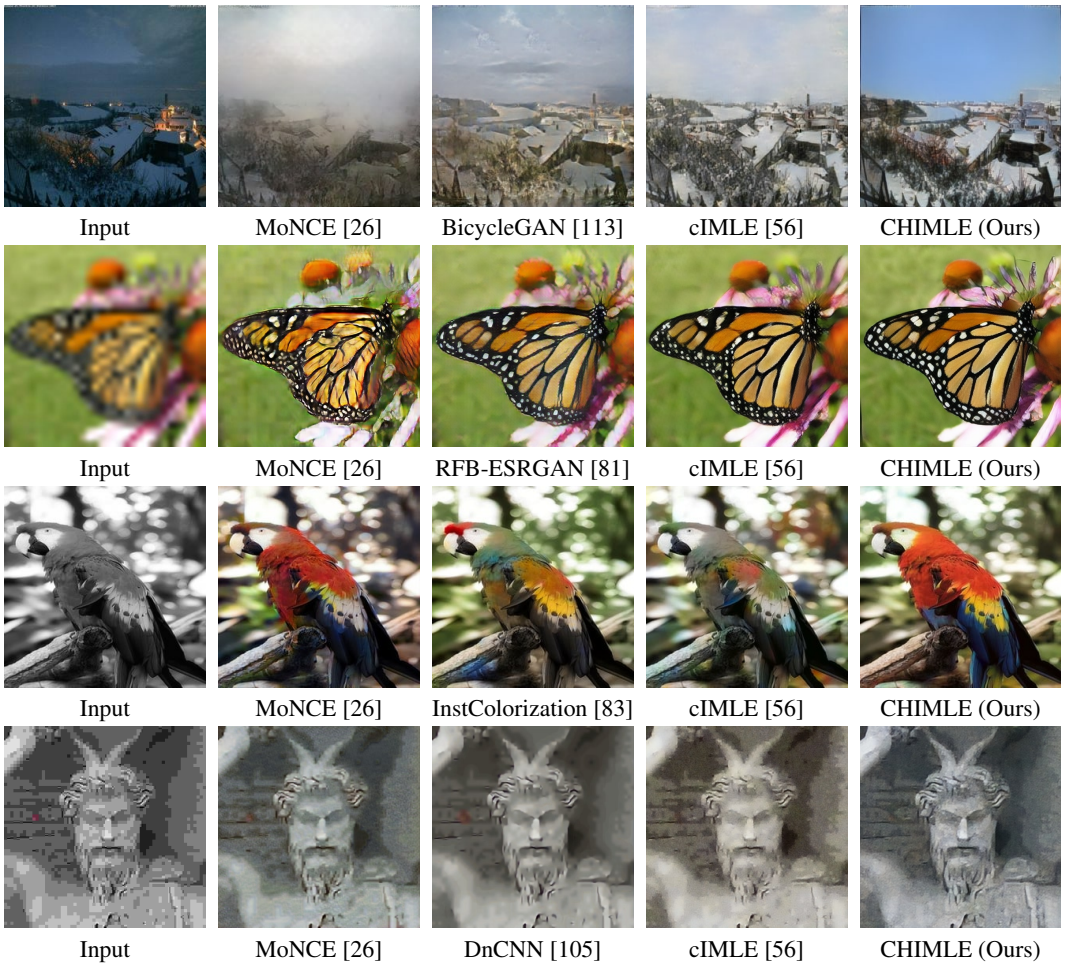


Figure 4: Qualitative comparison of our method (CHIMLE) and selected baselines on night-to-day (1st row), $16\times$ super-resolution (2nd row), image colourization (3rd row) and image decompression (4th row). In each row, we show the results from the most recent general-purpose baseline, MoNCE [26], a strong task-specific baseline (if applicable), the best prior IMLE-based method cIMLE [56] and CHIMLE. Better viewed zoomed-in. As shown, CHIMLE correctly preserves the details of the input while generating fine-grained details in the output, which compares favourably to the baselines.

comparison, we also compare our method to popular task-specific methods based on leaderboard rankings in challenges and recent survey papers [104, 4].

Tasks We apply our method to five different conditional image synthesis tasks, namely night-to-day, $16\times$ single image super-resolution, image colourization, image decompression and image synthesis from scene layouts. We show the first four in the main paper for space reasons. We pick night-to-day because it is a classic task in the multimodal general-purpose conditional image synthesis literature [38, 113]. We pick the remaining tasks because they are more challenging as evidenced by the availability of task-specific methods tailored to each task.

Training Details We use a four-module model introduced in Sect. 2.3 for all tasks. The input to each module is downsampled from the full resolution input to resolution the module operates on. We trained all models for 150 epochs with a mini-batch size of 1 using the Adam optimizer [46] with a learning rate of 1×10^{-4} on an NVIDIA V100 GPU. Details for datasets and PyTorch implementations are included in the supplementary materials.

Evaluation Metrics We evaluate perceptual quality of predictions using the Fréchet Inception Distance (FID) [31] and Kernel Inception Distance (KID) [6]. We also measure the diversity of the predictions since it is a crucial criterion of multimodal conditional image synthesis. One metric to measure diversity is the LPIPS diversity score (not to be confused with the LPIPS distance metric) [113], which is the average LPIPS distance between different output samples for the same

input. We observe that the LPIPS diversity score may favour a set of samples where some are of poor visual fidelity as shown in Figure 5. To get around this limitation, we use faithfulness-weighted variance (FwV) [56] instead, which is the average LPIPS distance d_{LPIPS} between the output samples and the mean, weighted by the consistency with the original image measured by a Gaussian kernel:

$$\mathcal{M} = \sum_i \sum_j w_i d_{\text{LPIPS}}(\tilde{\mathbf{y}}^{(i,j)}, \bar{\mathbf{y}}^{(j)}), \text{ where } w_i = \exp\left(\frac{-d_{\text{LPIPS}}(\tilde{\mathbf{y}}^{(i,j)}, \mathbf{y}^{(j)})}{2\sigma^2}\right) \tag{2}$$

where $\tilde{\mathbf{y}}^{(i,j)}$ is the ith generated sample from the jth input. $\bar{\mathbf{y}}^{(j)}$ is the mean of all generated samples from the jth input. $\mathbf{y}^{(j)}$ is the jth observed image. The kernel bandwidth parameter σ trades off the importance of consistency vs. diversity.

	LPIPS Diversity ↑	Faithfulness-weighted Variance (σ=0.2) ↑	Samples
Group 1	0.079	**0.0045**	
Group 2	**0.217**	0.0034	

Figure 5: Comparisons of LPIPS diversity score and faithfulness-weighted variance on two groups of sample images. Although Group 2 achieves a better LPIPS diversity score (shown on the left), some of the samples have poor visual fidelity. Group 1 contains samples that are both diverse (zoom in for difference in the seeds on the surface of the strawberry) and of high visual fidelity, which is preferred by the faithfulness-weighted variance (shown on the left).

We measure how well the modes match between the model distribution and the true distribution along two axes, mode accuracy (precision) and mode coverage (recall). We measure the former with the $F_{1/8}$ score component of the PRD metric [75], which is heavily weighted towards precision, and the latter with the F_8 score component of the PRD metric [75], which is heavily weighted towards recall, and the inference-via-optimization (IvO) metric [64], which measures the final mean squared error (MSE) between the generated image and the observed image after optimizing over the latent code input while keeping the model weights fixed.

3.1 Quantitative Results

We compare the perceptual quality and output diversity in Table 1. As shown, CHIMLE outperforms best prior IMLE-based method, cIMLE, general-purpose conditional image synthesis methods and task-specific methods in terms of FID and KID across all tasks. CHIMLE also outperforms all multimodal baselines in terms of FwV across all tasks. These results show that CHIMLE can produce more realistic and diverse images than the baselines.

We compare the mode accuracy and coverage in Table 2. As shown, CHIMLE outperforms or achieves comparable performance to cIMLE and other multimodal general-purpose baselines in terms of PRD scores. In addition, CHIMLE significantly outperforms cIMLE and other multimodal general-purpose baselines in terms of IvO. These results show that CHIMLE achieves better mode accuracy and coverage of the true distribution compared to the baselines.

3.2 Qualitative Results

We show the qualitative comparison of our method (CHIMLE) and selected baselines on various tasks in Figure 4. As shown, CHIMLE generates high-quality results, like the high contrasts in the building and the bushes in night-to-day (1st row), the sharp appearance of the wing pattern of the butterfly wing and the flowers in the background (2nd row), the vibrant yet detailed colouring of the parrot (3rd row) and the high-frequency details in the sculpture (4th row). Results for image synthesis from scene layouts and more comparisons to the baselines are included in the supplementary materials.

	Night-to-day			Super-Resolution (SR)			Colourization (Col)			Image Decompression (DC)		
	FID ↓	*KID* ↓	*FwV* ↑	*FID* ↓	*KID* ↓	*FwV* ↑	*FID* ↓	*KID* ↓	*FwV* ↑	*FID* ↓	*KID* ↓	*FwV* ↑
General-Purpose Methods:												
BicyleGAN [113]	179.08	132.35	0.23	67.17	53.28	0.84	53.33	34.44	7.44	87.35	20.24	0.29
MSGAN [63]	213.81	176.55	0.18	83.65	67.47	0.76	53.53	31.12	10.23	100.58	27.75	0.31
DivCo [59]	179.85	136.47	0.22	66.97	54.86	0.65	53.82	34.44	12.49	92.76	24.44	0.25
MoNCE [26]	232.10	217.79	0.18	47.97	35.31	0.68	27.67	9.55	18.30	113.21	27.56	1.12
Task-Specific Methods:												
RFB-ESRGAN (SR) [81]	–	–	–	19.90	8.01	–	–	–	–	–	–	–
InstColorization (Col) [83]	–	–	–	–	–	–	26.37	8.52	–	–	–	–
DnCNN (DC) [105]	–	–	–	–	–	–	–	–	–	109.38	48.05	–
IMLE-based Methods:												
cIMLE [56]	166.28	123.15	0.47	28.42	14.75	5.22	63.00	49.30	14.65	101.66	30.11	3.14
CHIMLE (Ours)	**141.44**	**90.10**	**0.51**	**16.01**	**4.54**	**5.61**	**24.33**	**7.73**	**27.11**	**73.69**	**12.12**	**3.80**

Table 1: Comparison of fidelity of generated images, measured by the Fréchet Inception Distance (FID) and Kernel Inception Distance (KID), and diversity, measured by faithfulness-weighted variance (FwV). We show the results by our method (CHIMLE) and the leading IMLE-based, task-specific baselines and general-purpose conditional image synthesis baselines. Lower values of FID/KID are better and a higher value of faithfulness-weighted variance shows more variation in the generated samples that are faithful to the observed image. For FwV, we show the results for the bandwidth parameter $\sigma = 0.2$ as it achieves a good balance between consistency and diversity. The KID and the FwV are shown on the scale of 10^{-3}. "$-$" indicates metric not applicable. We compare favourably relative to the baselines.

	Night-to-day			Super-Resolution (SR)			Colourization (Col)			Image Decompression (DC)		
	PRD ↑		*IvO* ↓	PRD ↑		*IvO* ↓	PRD ↑		*IvO* ↓	PRD ↑		*IvO* ↓
	F_8	$F_{1/8}$		F_8	$F_{1/8}$		F_8	$F_{1/8}$		F_8	$F_{1/8}$	
BicyleGAN [113]	0.95	**0.59**	61.10 ± 0.23	0.90	0.39	15.26 ± 0.38	0.94	0.63	11.85 ± 1.78	0.94	0.71	9.70 ± 2.93
MSGAN [63]	0.96	0.50	86.3 ± 2.03	0.92	0.42	16.51 ± 0.64	0.93	0.62	11.60 ± 2.07	**0.95**	0.74	8.00 ± 1.10
DivCo [59]	0.94	0.46	59.50 ± 0.91	0.87	0.33	13.82 ± 0.32	0.92	0.61	10.88 ± 1.20	0.93	0.68	6.38 ± 1.08
MoNCE [26]	0.94	0.43	19.40 ± 0.01	0.85	0.35	56.23 ± 0.71	0.95	0.75	13.10 ± 0.79	0.92	0.76	23.50 ± 0.10
cIMLE [56]	0.96	0.58	1.54 ± 1.04	0.86	0.32	6.78 ± 0.43	0.92	0.64	0.71 ± 0.09	0.91	0.74	5.86 ± 0.20
CHIMLE (Ours)	**0.97**	0.58	$\mathbf{0.96 \pm 0.04}$	**0.96**	**0.78**	$\mathbf{1.49 \pm 0.11}$	**0.97**	**0.87**	$\mathbf{0.32 \pm 0.01}$	**0.95**	**0.83**	$\mathbf{0.31 \pm 0.10}$

Table 2: Comparison of mode accuracy, measured by $F_{1/8}$ score of the PRD metric [75], and mode coverage, measured by F_8 score of the PRD metric [75] and the inference-via-optimization metric (IvO), between the model distribution and the true distribution by our method (CHIMLE) and the leading IMLE-based and general-purpose conditional image synthesis baselines. A higher value of the F_8 score shows better mode coverage (recall) and a higher value of the $F_{1/8}$ shows better mode accuracy (precision). A lower value of IvO shows better mode coverage since it indicate that the model can generate samples close to the observed images. For IvO, we show the average result across 5 independent runs on the scale of 10^{-3}. As shown, CHIMLE outperforms or achieves comparable performance to the baselines.

3.3 Ablation Study

We incrementally remove (1) iterative combination (IC), (2) partial evaluation (PE), (3) latent code division (CD). As shown in Figure 6, each component is critical to achieving the best results. Notably, we see an average increase in terms of FID by 7.7% after removing IC, by another 7.8% after removing PE and by another 12.8% after removing CD, thereby validating the importance of each of our contributions.

4 Related Work

General-Purpose Conditional Image Synthesis Conditional image synthesis can be divided into class conditioning [65, 68] where the output is conditioned on class labels, and image conditioning which takes an image as conditional input. The latter is often viewed as an image-to-image translation problem, which is most closely related to the setting we consider. General image-to-image translation methods can be paired or unpaired. In the paired setting, each image in the source domain corresponds to an image in the target domain [38, 102, 95, 80]. In the unpaired setting, images in the source domain may not correspond to images in the target domain [112, 101, 88, 58, 15, 7, 71]. Whereas these previous methods are all unimodal, other methods aim for the more challenging task of multimodal synthesis in the paired setting [113, 24, 21] and in the unpaired setting [34, 53, 54, 16]. Since many of the aforementioned works use GANs as their generators which suffer from mode collapse, some works aim to address this issue by introducing mode seeking terms [63] or adding contrastive losses [59, 26, 33]. In this paper, we focus on the paired multimodal setting and sidestep mode collapse by building on a recent generative modelling technique, conditional IMLE [56].

Full Model | No IC | No IC, PE | No IC, PE, CD

	SR	DC
Full Model	**16.01**	**73.69**
No IC	17.06	81.24
No IC, PE	18.53	87.95
No IC, PE, CD	21.13	101.38

Figure 6: Qualitative (left figure) and quantitative (right table) of Fréchet Inception Distance (FID) comparison as we gradually remove (1) iterative combination (IC), (2) partial evaluation (PE), (3) latent code division (CD) on two tasks, $16\times$ super-resolution (SR) (first row in the figure) and image decompression (DC) (second row in the figure). As shown in the figure, the generated image contains less fine details (as in SR, where the strawberry seeds become less pronounced) or colour saturation (as in DC) as we remove each component, which is also reflected by the FID shown in the table.

Task-Specific Approaches There is a large body of work on super-resolution, most of which consider upscaling factors of $2-4\times$. See [100, 66, 98] for comprehensive surveys. Many methods regress to the high-resolution image directly and differ widely in the architecture [22, 45, 50, 87, 28, 110, 18, 57]. Various conditional GANs have also been developed for the problem [52, 76, 97, 103, 70, 99, 96, 81]. For image colourization, a recent survey [4] provides a detailed overview of different methods. Some methods [13, 11, 108, 10] use deep networks to directly predict the colour and combine it with the input to produce the output. Other methods [37, 51, 19, 111, 83] take advantage of features from different levels or branches in the network for colour prediction. For image synthesis from scene layouts, some methods take input of a scene graph [30, 39] while others take a pixelwise semantic segmentation map as input [41, 86, 85, 84, 72]. There is relatively little work on image decompression to our knowledge. [5] targets DCT-based compression methods and explicitly models the quantization errors of DCT coefficients. More work was done on learned image compression [3, 2], which changes the encoding of the compressed image itself. Another related area is image denoising, see [89] for a survey of deep learning methods. Many of the methods uses a ResNet [29] backbone or other variants of deep CNNs [105, 106, 107, 94].

5 Discussion and Conclusion

Limitation Our method can only generate output images that are similar to some training images. Take colourization as an example, if there is no green apple in the training data, then our model cannot produce green apples.

Societal Impact CHIMLE could be applied to a broad range of conditional image synthesis tasks. While most of the applications are harmless, it could be potentially used for tasks like demosaicing which may constitute a privacy or copyright violation. In addition, since CHIMLE adopts a modular design and could include more modules to produce more refined results, training such huge model may increase greenhouse gas emissions.

Conclusion In this paper, we developed an improved method for the challenging problem of multimodal conditional image synthesis based on the IMLE framework. We addressed the undesirable tradeoff between sample efficiency and sample quality of the prior best IMLE-based method by introducing a novel hierarchical algorithm. We demonstrated on a wide range of tasks that the proposed method significantly improves output image quality without compromising diversity or mode coverage.

References

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

[2] 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*, pages 221–231, 2019.

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

[4] 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.

[5] Yuval Bahat and Tomer Michaeli. Explorable decoding of compressed images. *arXiv preprint arXiv:2006.09332*, 2020.

[6] Mikolaj Binkowski, Danica J. Sutherland, Michal Arbel, and Arthur Gretton. Demystifying mmd gans. *ArXiv*, abs/1801.01401, 2018.

[7] Konstantinos Bousmalis, N. Silberman, David Dohan, D. Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 95–104, 2017.

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

[9] Bingyi Cao, Andre F. de Araújo, and Jack Sim. Unifying deep local and global features for image search. In *ECCV*, 2020.

[10] Yun Cao, Zhiming Zhou, Weinan Zhang, and Yong Yu. Unsupervised diverse colorization via generative adversarial networks. *ArXiv*, abs/1702.06674, 2017.

[11] Fabio Maria Carlucci, Paolo Russo, and Barbara Caputo. $(de)^2$co: Deep depth colorization. *IEEE Robotics and Automation Letters*, 3:2386–2393, 2018.

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

[13] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 415–423, 2015.

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

[15] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018.

[16] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8185–8194, 2020.

[17] Bin Dai, Ziyu Wang, and David Paul Wipf. The usual suspects? reassessing blame for vae posterior collapse. In *ICML*, 2020.

[18] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11065–11074, 2019.

[19] Aditya Deshpande, Jiajun Lu, Mao-Chuang Yeh, Min Jin Chong, and David Alexander Forsyth. Learning diverse image colorization. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2877–2885, 2017.

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

[21] J. Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *ArXiv*, abs/1605.09782, 2017.

[22] 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*, pages 184–199. Springer, 2014.

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

[24] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martín Arjovsky, Olivier Mastropietro, and Aaron C. Courville. Adversarially learned inference. *ArXiv*, abs/1606.00704, 2017.

[25] Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub, Harm de Vries, Aaron C. Courville, and Yoshua Bengio. Feature-wise transformations. *Distill*, 2018.

[26] Yingchen Yu Rongliang Wu Shijian Lu Fangneng Zhan, Jiahui Zhang. Modulated contrast for versatile image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[27] 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*, pages 2672–2680, 2014.

[28] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018.

[29] 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)*, pages 770–778, 2016.

[30] Roei Herzig, Amir Bar, Huijuan Xu, Gal Chechik, Trevor Darrell, and A. Globerson. Learning canonical representations for scene graph to image generation. In *ECCV*, 2020.

[31] 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*, pages 6626–6637, 2017.

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

[33] Xueqi Hu, Xinyue Zhou, Qiusheng Huang, Zhengyi Shi, Li Sun, and Qingli Li. Qs-attn: Query-selected attention for contrastive learning in i2i translation. *arXiv preprint arXiv:2203.08483*, 2022.

[34] X. Huang, Ming-Yu Liu, Serge J. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. *ArXiv*, abs/1804.04732, 2018.

[35] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1510–1519, 2017.

[36] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

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

[38] 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*, pages 1125–1134, 2017.

[39] J. Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1219–1228, 2018.

[40] 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.

[41] Levent Karacan, Zeynep Akata, Aykut Erdem, and Erkut Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. *arXiv preprint arXiv:1612.00215*, 2016.

[42] 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)*, pages 4396–4405, 2019.

[43] 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)*, pages 8107–8116, 2020.

[44] 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*, pages 4401–4410, 2019.

[45] 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*, pages 1646–1654, 2016.

[46] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

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

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

[49] 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.

[50] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5835–5843, 2017.

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

[52] 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, page 4, 2017.

[53] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018.

[54] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Kumar Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision*, pages 1–16, 2020.

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

[56] 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.

[57] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3867–3876, 2019.

[58] Ming-Yu Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017.

[59] Rui Liu, Yixiao Ge, Ching Lam Choi, Xiaogang Wang, and Hongsheng Li. DivCo : Diverse conditional image synthesis via contrastive generative adversarial network. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16372–16381, 2021.

[60] James Lucas, G. Tucker, Roger B. Grosse, and Mohammad Norouzi. Don't blame the elbo! a linear vae perspective on posterior collapse. In *NeurIPS*, 2019.

[61] Zhiming Luo, Akshaya Kumar Mishra, Andrew Achkar, Justin A. Eichel, Shaozi Li, and Pierre-Marc Jodoin. Non-local deep features for salient object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6593–6601, 2017.

[62] Xuezhe Ma, Xiang Kong, Shanghang Zhang, and Eduard H. Hovy. Decoupling global and local representations via invertible generative flows. In *ICLR*, 2021.

[63] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1429–1437, 2019.

[64] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. In *ICLR*, 2017.

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

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

[67] 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.

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

[69] Yingxue Pang, Jianxin Lin, Tao Qin, and Zhibo Chen. Image-to-image translation: Methods and applications. *ArXiv*, abs/2101.08629, 2021.

[70] Seong-Jin Park, Hyeongseok Son, Sunghyun Cho, Ki-Sang Hong, and Seungyong Lee. SRfeat: Single image super-resolution with feature discrimination. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 439–455, 2018.

[71] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, 2020.

[72] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

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

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

[75] Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *NeurIPS*, 2018.

[76] 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)*, pages 4501–4510, 2017.

[77] 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.

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

[79] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017.

[80] Tamar Rott Shaham, Michaël Gharbi, Richard Zhang, Eli Shechtman, and Tomer Michaeli. Spatially-adaptive pixelwise networks for fast image translation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14877–14886, 2021.

[81] Taizhang Shang, Qiuju Dai, Shengchen Zhu, Tong Yang, and Yandong Guo. Perceptual extreme super resolution network with receptive field block. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1778–1787, 2020.

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

[83] Jheng-Wei Su, Hung kuo Chu, and Jia-Bin Huang. Instance-aware image colorization. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7965–7974, 2020.

[84] W. Sun and Tianfu Wu. Learning layout and style reconfigurable gans for controllable image synthesis. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2021.

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

[86] Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R. Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. *ArXiv*, abs/2003.07449, 2020.

[87] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3147–3155, 2017.

[88] Yaniv Taigman, A. Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *ArXiv*, abs/1611.02200, 2017.

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

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

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

[92] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *ArXiv*, abs/1601.06759, 2016.

[93] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Deep networks for saliency detection via local estimation and global search. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3183–3192, 2015.

[94] Tianyang Wang, M. Sun, and Kaoning Hu. Dilated residual network for image denoising. *ArXiv*, abs/1708.05473, 2017.

[95] 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*, pages 8798–8807, 2018.

[96] 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, 2018.

[97] 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*, pages 864–873, 2018.

[98] 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.

[99] Xiangyu Xu, Deqing Sun, Jinshan Pan, Yujin Zhang, Hanspeter Pfister, and Ming-Hsuan Yang. Learning to super-resolve blurry face and text images. In *Proceedings of the IEEE international conference on computer vision*, pages 251–260, 2017.

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

[101] Zili Yi, Hao Zhang, P. Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2868–2876, 2017.

[102] Donggeun Yoo, Namil Kim, S. Park, Anthony S. Paek, and In-So Kweon. Pixel-level domain transfer. *ArXiv*, abs/1603.07442, 2016.

[103] Yuan Yuan, Siyuan Liu, Jiawei Zhang, Yongbing Zhang, Chao Dong, and Liang Lin. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 701–710, 2018.

[104] 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)*, pages 2045–2057, 2020.

[105] 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.

[106] K. Zhang, W. Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27:4608–4622, 2018.

[107] Kai Zhang, W. Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3262–3271, 2018.

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

[109] 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*, pages 586–595, 2018.

[110] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. *CoRR*, abs/1802.08797, 2018.

[111] Jiaojiao Zhao, J. Han, Ling Shao, and Cees G. M. Snoek. Pixelated semantic colorization. *International Journal of Computer Vision*, 128:818–834, 2019.

[112] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.

[113] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *NIPS*, 2017.

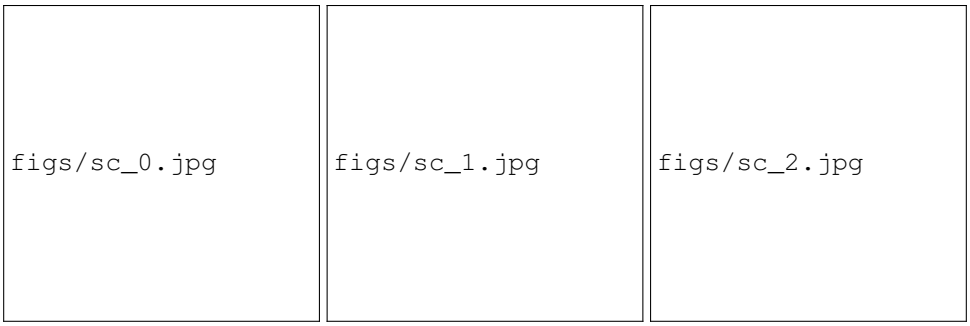
Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5 for details.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5 for details.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] No theoretical results.
   (b) Did you include complete proofs of all theoretical results? [N/A] No theoretical results.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplementary material for details.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 3 for details.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Existing assets are cited.
   (b) Did you mention the license of the assets? [N/A] Assets used in this research carry no license/access restrictions.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A] No new assets.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] No new assets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] No personally identifiable information in the data.
5. If you used crowdsourcing or conducted research with human subjects...

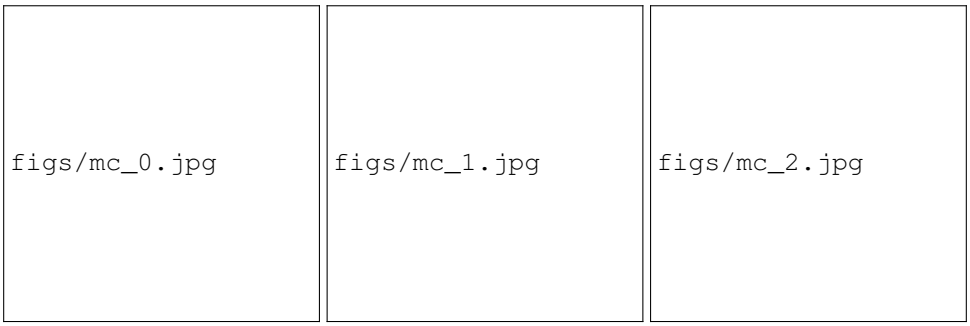
(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] No crowdsourcing.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] No crowdsourcing.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] No crowdsourcing.



Varying Latent Code To The Local Stream



Varying Latent Code To The Global Stream

Figure 7: Varying latent code input to (a) the local steam: capturing different local spatial features like the distribution of colour spots on the image. In this example, the wing and the breast of the parrot change from red to blue. (b) the global steam: capturing diverse global features like the colour tone of the image. In this example, the tone changes from red to blue and to green.

A Appendix

A.1 Implementation Details

Global and Local Stream Effects In Figure 7a we can see that by varying C_l only, local textures details like the distribution of the colour on the parrot's wing change while in Figure 7b, varying C_g only global features like the colour tone change.

Architecture In terms of the backbone architecture of the two streams, the local processing stream consists of residual-in-residual dense blocks (RRDB) [96]. Unlike traditional RRDB, which comes without normalization and deliberately removes batch normalization in particular, we apply weight normalization [78] to all convolution layers. The global processing stream consists of a sequence of dense layers, known as a mapping network [44]. The outputs from the mapping network apply adaptive instance normalization (AdaIN) [35, 25] to the output produced by each RRDB.

Details of RRDB In Figure 8, we show the inner workings of each RRDB, which contains dense blocks and residual connections.

Cross-Modular Dependency Adding a global stream increases the flexibility of each module. However, this introduces potential conflicts among global effects to the inputs of different modules. Figure 9 shows a concrete example of such a conflict where the first module's global stream colours the input red while the second module's global stream colours the input green. The final output from the modules becomes the undesirable gray colour due to the conflicting effects of the two global streams. To counter these conflicts, we propose to add a cross-modular dependency among different modules' global streams with two variants, sequential dependency and dense dependency. As shown in Figure 10, the sequential dependency scheme connects the global streams of neighbouring modules,

figs/RRDB.jpg

Residual-in-Residual Dense Block (RRDB)

figs/RDB.jpg

Dense Block

Figure 8: (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.

and the dense dependency scheme connects each global stream to all previous modules. Our final model adopts the dense dependency scheme since it performs the best, as later shown in Sect. 3.3.

figs/dependency_example.jpg

Figure 9: Conflict scenario for the vanilla architecture with no dependency among global streams. If the first module's global stream has the effect of colouring the input red, and the second module's global stream has the effect of colouring the input green. Then the output from the two modules would be the unwanted gray colour.

figs/no_dep.jpg	figs/seq_dep.jpg	figs/dense_dep.jpg
No Dependency	Sequential Dependency	Dense Dependency

Figure 10: Alternatives for global stream dependencies among different modules. (a) Vanilla architecture: no dependencies among global streams. (b) Sequential dependency scheme: connects the global streams in neighbouring modules. (c) Dense dependency scheme: connects each global stream to all previous modules.