# InvNet: Encoding Geometric and Statistical Invariances in Deep Generative Models

**Ameya Joshi**[*1], **Minsu Cho**[*1], **Viraj Shah**[1], **Balaji Pokuri**[2], **Soumik Sarkar**[2], **Baskar Ganapathysubramanian**[2], **Chinmay Hegde**[1]
[1] Dept. of Electrical and Computer Engineering [2] Dept. of Mechanical Engineering
Iowa State University, Ames
ameya, chomd90, viraj, balajip, soumiks, baskarg, chinmay@iastate.edu

## Abstract

Generative Adversarial Networks (GANs), while widely successful in modeling complex data distributions, have not yet been sufficiently leveraged in scientific computing and design. Reasons for this include the lack of flexibility of GANs to represent discrete-valued image data, as well as the lack of control over physical properties of generated samples. We propose a new conditional generative modeling approach (InvNet) that efficiently enables modeling discrete-valued images, while allowing control over their parameterized geometric and statistical properties. We evaluate our approach on several synthetic and real world problems: navigating manifolds of geometric shapes with desired sizes; generation of binary two-phase materials; and the (challenging) problem of generating multi-orientation polycrystalline microstructures.

## 1 Introduction

**Motivation.** Generative Adversarial Networks (GANs) have proven to be highly successful in synthesizing samples arising from complex distributions, including face images (Radford, Metz, and Chintala 2016), content generation (Jin et al. 2017), image translation (Isola et al. 2017), style transfer (Zhu et al. 2017), and many others. Our motivation for this paper arises from computational engineering design, for which promising progress has been made in areas such as drug discovery (Blaschke et al. 2018), molecule design (Sanchez-Lengeling and Aspuru-Guzik 2018), and 3D modeling (Wu et al. 2016).

Computational design problems often are accompanied by stringent geometric and statistical constraints that all valid solutions are required to satisfy. These constraints are generally informed by the physics of the problem, or by manufacturing limitations. Additionally, the solution spaces for several design problems are often discrete (integer) valued, often combinatorially complex, and generally non-differentiable, therefore disallowing the use of scalable traditional optimization tools. Machine learning methods such as GANs show the promise of sidestepping some of these concerns. However, there remain several challenges that must be overcome for realistic design problems.

**Challenges**. When standard GAN models are applied to solve real-world design problems, several challenges arise. It is well known that GANs incur dramatically *high sample complexity*. For example, BigGAN (Brock, Donahue, and Simonyan 2019) requires 14 million (natural) images trained over $\sim$24K TPU-hours, which is well beyond the reach of normal computing environments. This cost is exacerbated in scientific and engineering design problems that rely on expensive simulations for training data generation. A possible solution is to use *a priori* domain knowledge about the *physics* of the design problem to reduce training data requirements, but the standard GAN framework does not leverage such knowledge.

Further, designers often require fine-grained input control over specific parameters of the search space. For example, in (computational) materials design, the designer may wish to adjust material composition, grain sizes, or other material property parameters on the fly. Conditional GANs (Odena, Olah, and Shlens 2017; Mirza and Osindero 2014) do provide some amount of input control, but these are coarse: the training data has to be carefully binned due to the categorical conditioning involved.

**Our contributions**. We introduce *InvNet*, an architecture that extends deep generative models (such as GANs) by encoding user-specified geometric (discrete) and statistical constraints. Our InvNet generative model has the dual advantage of being able to learn *implicit* features from training data, while also being able to *explicit* user-specified invariances. Similar to GANs, we pose the InvNet training problem as a minimax game and propose a three-way alternating-optimization style training algorithm. Our algorithm requires minimal parameter tuning and gives stable results across a wide range of problem domains.

We showcase our framework in the context of two challenging problems in materials informatics. In both cases, the physics governing the formation of such microstructures are typically very complex and involve solving nonlinear high-order partial differential equations (PDEs) that are computationally very intensive. In both problems, we show that InvNet is successful in generating a large, diverse variety

[*]Equal contribution

of material microstructure samples that respect the specified invariances, as well as enables flexible user navigation of the design space.

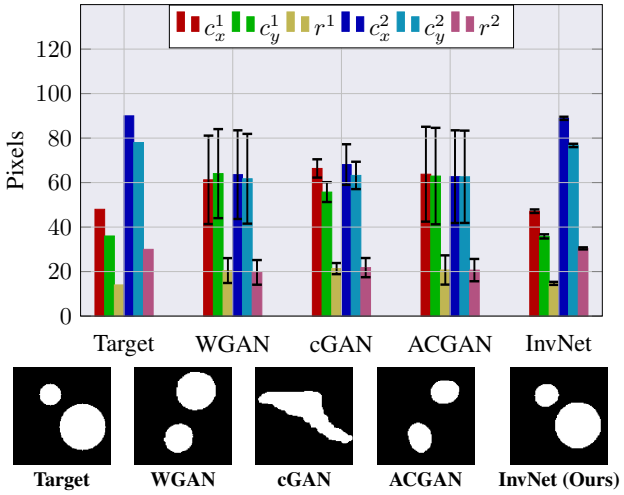



Figure 1: *Examples from the manifold of images of two circles (with user-specified positions and radii) synthesizes using various generative models. The bar/whisker plots show the accuracy of each model for respecting the specified invariances (center positions and radii for the $i^{th}$ circle are represented by $(c_x^i, c_y^i)$ and $r^i$ respectively). WGAN provides no explicit control over the parameters of the generated circles, while cGAN and AC-GAN fail to generate satisfactory images. Our proposed InvNet model not only learns to reproduce correct shapes, but also provides effective user control over the location and size of the shapes. The large standard deviation for the properties show that the other approaches fail to learn the associated invariances.*

Our specific contributions are as follows:

1. We describe InvNet, a novel generative modeling framework which respects user-defined geometric and/or statistical invariances.
2. We present a rigorous experimental analysis for the synthetic example of generating composite images of shapes with user-controlled positions and sizes. We also show superior performance over related modeling methods such as conditional GANs.
3. We demonstrate the efficacy of InvNet for two challenging real-world applications in material discovery: (1) generating binary microstructures with specified statistical properties; (2) poly-crystalline (metal alloy) microstructures with specified geometrical and statistical properties.

**Paper outline.** We discuss relevant literature for generative models in general and for the specific case of computational design in Sec. 2. We then describe our approach in detail in section Sec. 3. Sec. 4 presents an illustrative example for InvNet where we train a generative model to generate simple shapes with specified geometric constraints such as position and size. Subsequently, we present two real world applications in materials design and discovery; for two phase and multi-orientation poly-crystalline materials respectively in Sec. 5 and Sec. 6. Finally we conclude with a brief discussion in Sec. 7.

## 2 Related Work

Due to tight page-limit constraints, we defer a full discussion of related work to the appendix.

**Conditional GANs.** Conditional GANs(cGANs) (Mirza and Osindero 2014) and AC-GANs (Odena, Olah, and Shlens 2017) condition the generator on categorical labels to control the output class by modifying the discriminator architecture, while InfoGAN (Chen et al. 2016) permits control over output parameters through an information-maximization loss term. While the goal of our InvNet model is thematically similar, InvNet is more flexible in that we enable fine-grained (continuous) control of the output. To achieve this, InvNet requires important architectural choices: as opposed to cGANs where the discriminator is modified, we use a specific closed-form *invariance checker* in addition to a standard GAN discriminator. This two-pronged setup reflects building an associative mapping for the invariance while also discriminatively learning other features from training data. We elaborate in Sec. 3.

The approach of (Stinis et al. 2018) employs a noisy data-training approach with mathematical constraints in order to extrapolate the generator distribution. While they use the approach of weakening the discriminator with noisy inputs, we use an alternating optimization scheme to encode invariances. Additionally, our architecture is extensible to more general geometric and discrete constraints. Finally, (Jiang et al. 2019) use segmentation masks to enforce structural constraints; this resembles our geometric constraint modelling.

**Materials design.** An entire sub-field in computational material science is devoted for the synthetic generation of material microstructures (Ganapathysubramanian and Zabaras 2008; 2007; Roberts 1997). Examples of synthesis methods include Gaussian random fields (Roberts 1997), optimization-based methods (Yeong and Torquato 1998), and, multi-point statistics (Feng et al. 2018). Recent advances also involve the generative modeling techniques (Sanchez-Lengeling and Aspuru-Guzik 2018) that largely rely on the massive training datasets. Most of these methods involve large scale expensive physics simulations to generate a massive number of candidate designs, followed by rejection-sampling to choose the desired solutions. On the other hand, InvNet is directly trained to enforce statistical and geometric constraints while also permitting user-defined exploration of the solution space.

## 3 The InvNet Model

Consider a data distribution $\mathbb{P}_{\text{data}}$ defined over a set $\mathcal{D} \subseteq \mathbb{R}^d$, and a list of differentiable *invariance* functions $\mathbb{R}^d \rightarrow \mathbb{R} : I_i(\cdot),\ i = 1, 2, .., r$. The aim of InvNet is to generate new samples $\mathbf{x}$ from $\mathcal{D}$ that satisfy an invariance, $I_i(\mathbf{x}) = 0,\ \forall i = 1, 2, .., r$. We define our generator to be a function $G_\theta : \mathbb{R}^k \rightarrow \mathbb{R}^d$ parameterized by $\theta$. Let $\mathbf{z}$ represent a $k$-dimensional latent input vector to the generator.

In the standard GAN setup (Goodfellow et al. 2014), the generator is trained by posing a two-player game between the generator ($G$) and the discriminator ($D$), where the discriminator is a function $D_\psi : \mathbb{R}^d \rightarrow \mathbb{R}$ parameterized by $\psi$.

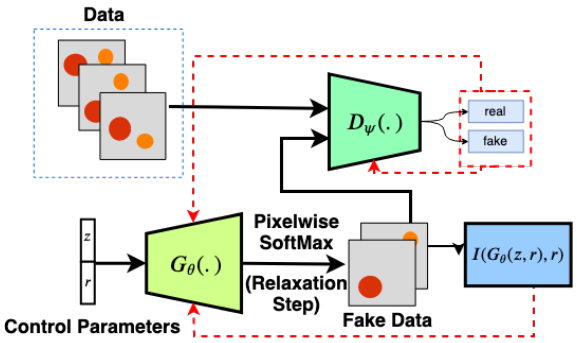



Figure 2: *Our proposed InvNet model introduces an invariance function ($I(\cdot)$) along with the traditional generator ($G$) and discriminator ($D_\psi$). While the discriminator learns the implicit features of the image through simultaneous training of both $D$ and $G$, the invariance enforces a statistical/geometric constraint on the generator ($G_\theta$) through minimizing an invariance loss. InvNet handles discrete (integer) valued invariances by relaxing the integer valued data into a probabilistic space using pixelwise softmax activations.*

The training objective of GANs is given by:

$$L(\theta,\psi) = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{data}}}\left[f(D_\psi(\mathbf{x}))\right] + \mathbb{E}_{\mathbf{z}\sim\mathbb{P}_z}\left[f(-D_\psi(G_\theta(\mathbf{z})))\right], \tag{1}$$

for some monotonic function $f : \mathbb{R} \to \mathbb{R}$ and $\mathbb{P}_z$ being a known distribution. We focus on Wasserstein GAN (Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017) where $f(t) = t$.

In order to encode invariances, we propose solving the following minimax game which will produce *Invariance Networks* (InvNets):

$$\begin{aligned} \min_\theta \max_\psi \quad & L(\theta,\psi) + \mu L_I(\theta) \\ \text{where } L_I(\theta) \quad & := \textstyle\sum_{i=1}^n \mathbb{E}_{\mathbf{z}}\left[I_i(G_\theta(\mathbf{z}))\right] \end{aligned} \tag{2}$$

We solve the minimax game in a fashion similar to GAN training; we alternately adjust the generator parameters $\theta$ and the discriminator parameters $\psi$ via gradient updates. However, due to the presence of the additional invariance term in $\bar{L}(\theta,\psi)$, we find in practice that a three-way update rule works well: a GAN-like update of $\theta$ via gradient steps of $L(\theta,\psi)$ keeping $\psi$ fixed; a GAN-like update of $\psi$ via gradient steps of $L(\theta,\psi)$ keeping $\theta$ fixed; and an update of $\theta$ via gradient steps of $L_I$. See Alg 1.

**Role of the input vector.** We define the generator input as the concatenated vector $[\mathbf{z},\mathbf{r}]^T$ with $\mathbf{z}$ referring to a random vector sampled from a known distribution while $\mathbf{r}$ parameterizes the invariances. Specifically, $\mathbf{r}$ is a vector that corresponds to *tunable* geometric or statistical parameters of the generated data; thus allowing for user control. For e.g, for the case of generating a set of circles on a plain background, $\mathbf{r}$ would be a concatenated vector of radii and centers of the respective circle.

**Differences from Conditional GANs.** While noting that InvNet has a few similarities to conditional GAN models such as cGANs (Mirza and Osindero 2014) and AC-GANs (Odena, Olah, and Shlens 2017), we list the major differences and advantages.

In particular, while cGANs (Mirza and Osindero 2014) surrogate the conditional distribution, $\mathbb{P}_{\mathbf{x}|y}$ by passing both the data and label to the discriminator, the discriminator in InvNet learns to model the true data distribution, $\mathbb{P}_{\mathbf{x}}$ unbiased by the constraints.

Conversely, AC-GAN modifies the discriminator to perform the simultaneous task of discrimination and classification or regression. InvNets are essentially simplifications of this idea. The invariance function, $I_i(.)$ can be thought of as a fixed auxiliary model, that is decoupled from the discrimination task. This allows the discriminator to be less constrained in comparison, thus forcing the generator to learn the true data distribution better. Additionally, due to the closed form representation of the fixed auxiliary loss, the data requirements in terms of variety is reduced.

**Variations of alternating optimization.** As mentioned above, we optimize our multi-objective formulation by alternately optimizing over the three sub-components of $\bar{L}$, as presented in Alg. 1. Our choice of using alternating optimization is informed by insights in recent work (Mokhtari, Ozdaglar, and Pattathil 2019) that show that regular gradient descent for minimax games diverges, while methods that take intermediate gradient steps, such as *extra-gradient descent*, converge to stable Nash equilibria.

**Algorithm 1** Training InvNets

**Require:** Set learning rates, termination conditions; Training data: $\mathbf{x} \sim [c]^d$.

**while** $L_I$ large and $\theta$ has not converged **do**
  **for** $l \leftarrow 1$ to $N_G$ **do**
    $\theta \leftarrow \theta - \eta_G \nabla_\theta \bar{L}(G_\theta(\mathbf{z},\mathbf{r}))$ ▷ Generator update
  **end for**
  **for** $m \leftarrow 1$ to $N_D$ **do**
    $\psi \leftarrow \psi + \eta_D \nabla_\psi \bar{L}(\bar{\mathbf{x}})$ ▷ Discriminator update
  **end for**
  **for** $n \leftarrow 1$ to $N_I$ **do**
    $\theta \leftarrow \theta - \eta_D \nabla_\theta L_I$ ▷ Projection step
  **end for**
**end while**

## 4 Toy Example: Generating shapes with geometric constraints

We start with the stylized problem of learning image manifolds, where the training data consists of simple shapes parameterized by geometrical quantities such as size and position. Here, we aim to train an InvNet that generates shapes with user-specified sizes and/or positions.

Consider the illustrative problem of generating *two circles* of varying sizes and positions on a plain background. While traditional GANs can generate such shapes by learning from data, suppose that we additionally require *independent* control over the radius and position of each of these circles.

Let us first consider how this can be achieved for a single circle. A binary image with a white circle satisfies two invariances: (1) the area occupied by the white pixels should

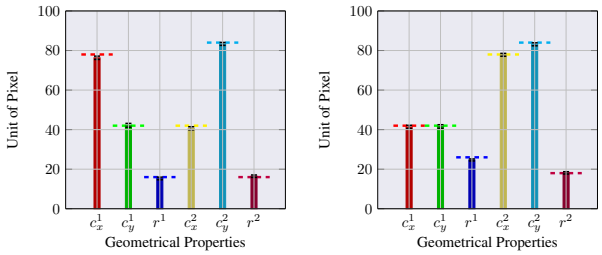



Figure 3: *Performance of InvNet in respecting target invariances. The bar plots show the centroids* $(c_x^i, c_y^i)$ *and radii,* $r^i$ *for images generated by InvNet for the* two circle *task. Note that InvNet successfully generates images that respect the required target invariance value (represented by the dashed horizontal line) with very low error.*

be $\pi$ times the radius squared; and (2) the position of the circle center is the center of mass of a single, connected component. The first invariance can be easily expressed in the form of a continuous, differentiable loss function:

$$L_I^{\text{area}}(\mathbf{x}, r) = \left| \sum_i \mathbf{x}_i - \pi r^2 \right| \tag{3}$$

where $r$ is the target radius.

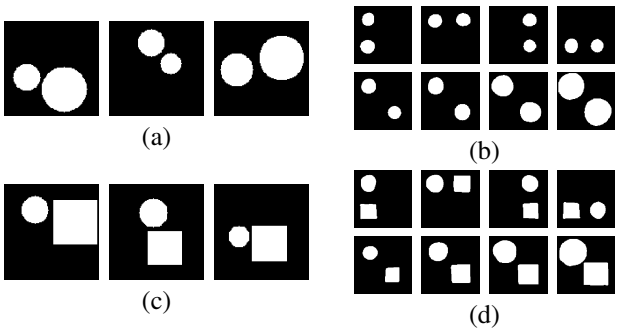



Figure 4: *Generated sample with multiple shapes. (a) & (c) Original datasets. The first dataset consists of images with two circles while the second consists of images with a circle and a square. (b) Generated images for the two-circle dataset. InvNet is successfully trained to generate circles of target radii and position. (d) Similarly, another example of generating simple shapes with a required area and position. The invariance function forces the generator to learn the mapping between the input condition and the target property for each connected component. Refer appendix Fig. 11, Fig. 12 for additional results.*

The second invariance, encoding the centroid of the circle requires the use of central moments, which is a well-known mechanism in image processing to calculate the center of mass of a connected component.

$$(c_x, c_y) = \left( \frac{\mu_{1,0}(\mathbf{x})}{\mu_{0,0}(\mathbf{x})}, \frac{\mu_{0,1}(\mathbf{x})}{\mu_{0,0}(\mathbf{x})} \right) \tag{4}$$

$$\mu_{m,n}(\mathbf{x}) = \sum_{k=1}^{w} \sum_{l=1}^{h} k^m l^n \mathbf{x}_{kh+l} \tag{5}$$

We use this to construct an invariance function that minimizes the $\ell_2$-error between the target centroid and the calculated centroid of the generated image.

$$L_I^{\text{pos}}(\mathbf{x}, c_x, c_y) = \left\| \begin{bmatrix} \mu_{1,0}(\mathbf{x})/\mu_{0,0}(\mathbf{x}) \\ \mu_{0,1}(\mathbf{x})/\mu_{0,0}(\mathbf{x}) \end{bmatrix} - \begin{bmatrix} c_x \\ c_y \end{bmatrix} \right\|_2^2 \tag{6}$$

Using Eq. 3 and Eq. 6 as invariances, it is straightforward to train InvNet to generate a *single* connected component (such as a circle) while controlling the radii and positions of the circle. The problem, however, becomes non-trivially challenging when *multiple* such connected components are involved, since the closed form expressions in Eq. 3 and 6 only provide meaningful information only when one geometrical figure exists in the image. We are not aware of (differentiable) invariance functions which calculate similar geometric quantities for the case of multiple connected components.

To overcome this challenge, we use the following intuition. The discrete constraint of having two connected components can be treated as *a coloring* (or assignment) problem, where each pixel can be assigned to one of three possibilities: either of the connected components, or neither of them (background). Therefore, this intuition motivates us to relax the assignment constraint by training a generator *with three color channels*, each corresponding to one of the assignment categories.

Formally, we describe the setup as follows. Consider a dataset, $\mathbf{x} \sim [c]^d$ where each pixel can take a value in $\{0, 1, \ldots, c\}$. For the case described above, the pixel value defines the class of that pixel; whether it lies in any of the $c$ circles, or the background. In order to train InvNet to generate such images while enforcing the area and the positional invariance, we propose two modifications. The first is to encode the input (training) dataset into pixelwise one-hot representation i.e. $\bar{\mathbf{x}} \in \{0, 1\}^{(c+1)\times d}$. Secondly, the generator, $G(.)$ is now forced to generate $(c + 1)$ channels in its final layer. Consequently, in order to encode the assignment problem, we also add a pixelwise softmax activation to the final layer; allowing us to relax the discrete generation problem to a continuous bounded space, $[0, 1]^{(c+1)\times d}$.

The problem now decomposes to that of generating a single connected component in each channel, with the specified invariances according to Eq. 3 and Eq. 6 defined independently for each channel as follows:

$$L_I(G_\theta(\mathbf{z}, \mathbf{r})) = \sum_{i=1}^{c} L_I^{\text{area}}(G_\theta(\mathbf{z}, \mathbf{r})_i, \mathbf{r}_i) + L_I^{\text{pos}}(G_\theta(\mathbf{z}, \mathbf{r})_i, \mathbf{r}_i) \tag{7}$$

where subscript refers to the index of the channel.

Following this approach, we train an InvNet with generator, $G(\mathbf{z}, \mathbf{r})$, that takes a tuple of the two radii and the centroids respectively as input. The dataset consists of two non-overlapping circles with varying radii, $r \in \{14, 15, \ldots, 32\}$ and centroids, $(c_x, c_y) \in \{30, 31, \ldots, 96\}^2$ (a unit of pixel) sampled uniformly to avoid bias. We discard any samples if any two figures overlap each other to obtain datasets with non-overlapping geometrical figures. Additionally, clusters of pixels with label 1 and 2 form two different figures while

remaining pixels are assigned to label 3 as a background of the image. The training data consists of $30k$ images of size $128 \times 128$.

During training, $\mathbf{x}$ is one-hot encoded so that the generator generates a tensor of $128 \times 128 \times 3$. The discriminator attempts to the generated pixelwise softmax distribution to the input. The generator, discriminator, and the invariance objectives are then optimized using the three-way alternating optimization method described. An important point to note here is that while the invariance losses explicitly encode geometric properties of size and position, the shape is learnt from the underlying data. Examples of generated images can be seen in Fig. 4.

We also show that InvNet can also be used to learn other shapes from data while still enforcing geometric constraints. For the second example, we consider a dataset of a circle and a square on a plain background. We subsequently modify the invariance functions appropriately. Note, however that in this case, we input the target area of each shape so as to not bias the generator towards either shape. The InvNet is successful in learning to generate the two shapes (refer Fig. 4.). We refer the reader to the appendix (Fig. 11 and Fig. 12) for additional results.

We also analyse the efficacy of the model in encoding the target invariances. We generate $1000$ images for each instance for a varied set of target radii and positions for the two circle generation problem. As the results in Fig. 3 suggest, InvNet is able to generate a large variety of examples while accurately respecting the invariances. Additional results can be found in the appendix (Fig. 17).

**Comparisons.** While InvNet does show promising performance as described above, a question may arise about the use of an auxiliary loss as compared to using conditional GANs (Mirza and Osindero 2014; Odena, Olah, and Shlens 2017) that use surrogate neural networks for similar tasks. We, therefore, conduct a comparative study where we train a WGAN, a cGAN and an ACGAN to generate multiple circles with a required target radius and position. For fair comparisons we train all models with the same generator and for the same number of iterations.

We observe that InvNet successfully generates circles with the required constraints. On the other hand, the cGAN fails to capture the data distribution whereas the ACGAN fails to generate images satisfying the required conditions. Fig. 1 shows a detailed analysis for a single target orientation. Additionally, InvNet converges faster than AC-GANs ($\sim$ $3k$ iterations to $\sim$ $23k$ iterations) for generating acceptable circles. Additional analysis can be found in the appendix.

## 5 Example: Two-phase microstructures

In computational material science, material distribution is represented by an image describing the arrangement of constituents within a material, whose statistics govern the physical properties of the underlying material. Synthesizing microstructures adhering to specific statistical properties is, therefore, a crucial component of material discovery.

We focus on binary microstructures (corresponding to black/white images) corresponding to two fluid constituents.

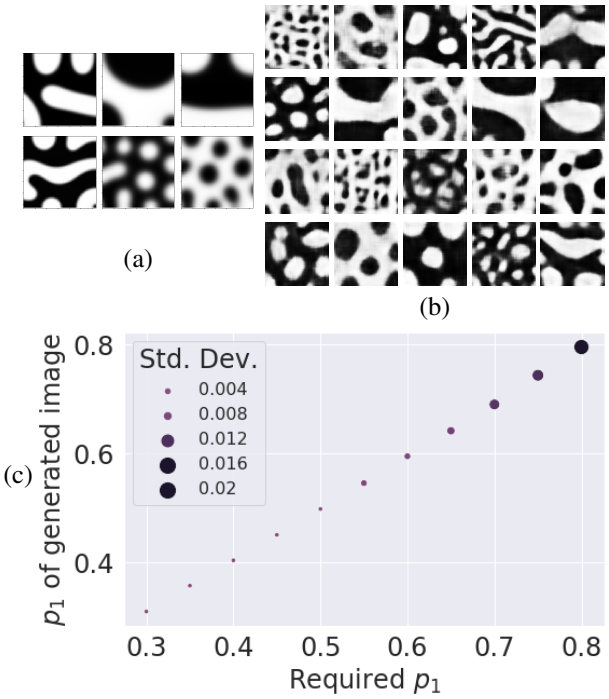



Figure 5: *Binary microstructures. (a) Original Dataset. (b) Generated microstructures. Each column is generated according to a specific required volume fraction. (c) Bubble plot for 1000 images for each input $p_1$ value. The size represents the standard deviation. Note that the error in volume fraction is less than 5% in all cases.*

Generally, the first and second moments of the image are useful statistical descriptions. Formally, we consider: (i) the $1^{st}$ moment, $p_1$, also called the *volume fraction*, and (ii) the $2^{nd}$ moment, $p_2$, also called the *two-point correlation*. The former is a scalar, while the latter is a function. The statistical moments are highly correlated with material properties such as thermal conductivity and elastic behaviour.

The dynamics of binary microstructures exhibiting *phase separation* are governed via the well-known Cahn-Hilliard (CH) equation (Cahn and Hilliard 1958). This is a fourth-order nonlinear PDE, and its solution requires a significant amount of simulation time (see Table 1). Therefore, synthesizing two-phase binary microstructures is computationally very challenging. We remedy this by training an InvNet to generate microstructures adhering to desired statistical properties. We consider two specific modes in this case; (1) generating microstructures with a required volume fraction, (2) generating microstructures with a required; $p_2$ correlation curve. Each of the two statistical parameters inform various material properties which can then be further analysed.

For generating microstructures with specific statistical properties, the generator $G(.)$ takes as input the latent vector, $\mathbf{z} \sim N(0, \mathbf{I})$ and the corresponding parameters; $\mathbf{r}$ (either $p_1$ or $p_2$). Since the first and second moments are differentiable functions, we encode the desired statistical properties into the InvNet formulation using the invariances:

$$L_I = \|I_{p_r}\left(G_\theta(\mathbf{z}, p_r)\right) - p_r\|_2^2 \tag{8}$$

where $I_{p_r}$ represent the functional forms of the moments, and $p_r$ are the two moments appropriately.

For training the InvNet, we use a publicly available dataset of 2D binary microstructures containing $\sim 34k$ images across the wide range of statistical moments (Pokuri et al. 2019) (refer appendix for details). We train the InvNet using Alg. 1.

The results in Fig. 5 show the generated images adhering to target invariances. In order to analyse the efficacy of InvNet, we generate 500 images for specific values of moments; and calculate the mean and standard deviation of the properties of the generated images. Fig. 5(c) show the distribution of the generated moments for a range of values. Observe that InvNet is able to successfully generate microstructures with very low error in terms of $p_1$. We present additional results for $p_2$ in the appendix Fig. 13.

**Comparisons with other methods** We also compare with the approach proposed by Stinis et al. that enforces physically valid constraints by training the discriminator on data and the corresponding residual of the constraints. To ensure fair comparison, we extend the same concept to WGAN-GP. For real data, the value of the invariance function (Eq. 8) goes to 0. The discriminator subsequently, is input the tuple, $(\mathbf{x}, L_I(\mathbf{x}))$ for both the real and fake data during training. We observe that while InvNet successfully learns to generate examples similar to the training data that satisfies the required invariance, the GAN trained as above fails to converge with the discriminator loss exploding to a high $(\sim -10^7)$ value (Refer Fig. 6).

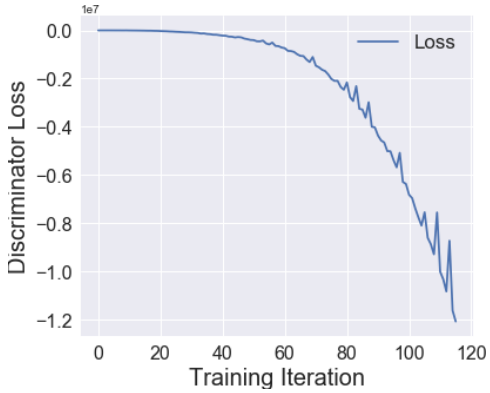



Figure 6: *Discriminator loss for the approach presented in Stinis et al.. Note that the discriminator loss explodes to very high values within a few iterations.*

In order to motivate the advantages of InvNet over traditional methods, we compare the time required to generate microstructures for our approach and the state of the art numerical method (Wodo and Ganapathysubramanian 2012). We note that the training time incurred by InvNet is amortized over the generation time required to simulate many microstructures; to generate 100,000 microstructures InvNet already obtains speedups over existing numerical solutions. The results of the comparison are presented in Table. 1. This computational advantage will scale with the number of candidate images required.

Table 1: *(Left) InvNet generated microstructures for fixed $1^{st}$ and $2^{nd}$ moments. (Right) Comparison with simulation times for generating $10^5$ microstructure images using numerical methods (Wodo and Ganapathysubramanian 2012).*

| Model type | Time (s) |
|---|---|
| Numerical solution (Wodo and Ganapathysubramanian 2012) | |
| Generating 1 microstructure† | $1.84s$ |
| Total time for 100000 images† | **184000s** |
| **InvNet** (our approach) | |
| Training time× | $57600s$ |
| Generating 1 microstructure× | $0.0110s$ |
| Total time for 100000 images× | **58700s** |

† Uses Intel 4-core CPU with 32 GB RAM.
× Uses 1 NVIDIA Tesla V100 GPU, 32 GB GDDR5 on TensorFlow GPU version 1.4.

## 6 Example: Polycrystalline Microstructures

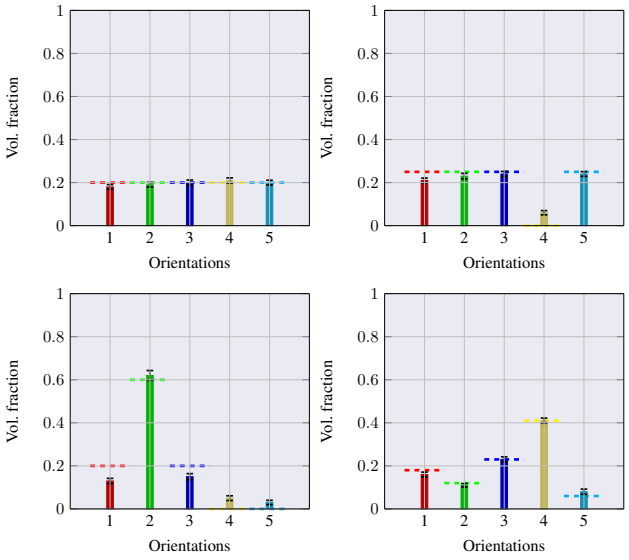
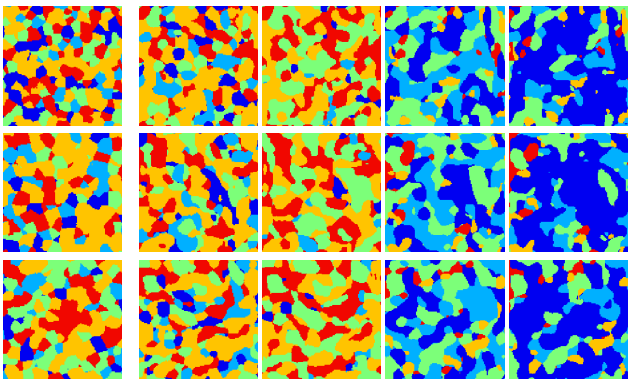



Figure 7: *Grain distribution for generated polycrystalline data. The plots represent the mean volume fractions of* 100 *images for each required distribution. The horizontal* dashed *lines denote the target volume fraction for the orientation. Observe that InvNet is successful at generating polycrystalline images for a large variety of grain distributions. Also note that several of the target distributions are not present in the training dataset.*

We now consider the challenging problem of generating microstructure images of polycrystalline materials, such as metal alloys. Polycrystalline materials consist of several small non-overlapping regions called *grains*. The orientation of atoms is necessarily different for adjoining grains. An example microstructure can be seen in Fig. 8 where each orientation is represented by a different color. The distribution of these regions correlates with mechanical and physical properties.

There are two difficulties in synthesizing polycrystalline microstructure images: (1) every grain in the image must be assigned to a specified orientation and, (2) each orientation must constitute a specific *volume fraction*. The volume frac-

tion is a statistical invariance, whereas the orientation constraint can be construed as a geometric invariance.



(a) (b)

Figure 8: *(a) Example microstructures from the polycrystalline dataset. Different orientations are represented by different colors. (b) Generated microstructures. Each column is generated according to a specific required volume fraction. InvNet is successfully able to learn to generate images corresponding to various target volume fractions.*

**Dataset.** For this specific example, we consider a dataset of microstructure images of a metal alloy generated using Dream3D (Groeber and Jackson 2014). Each image in the dataset consists of a distribution of five *orientations*. For the purposes of training, each orientation is mapped to a scalar integer value from $\{1, \cdots, 5\}$. Due to computational limitations, we scale the images down from the original $400 \times 400$ domain size to $128 \times 128$.

Given that the polycrystalline microstructure manifold is integer valued, optimization of the InvNet objective would not be feasible. Therefore, we use the approach defined in Sec. 4 to relax the integer valued assignment problem to a continuous valued probability space. For training, we encode the discrete microstructure map from the given dataset, $\mathbf{x} \in [c]^d; c \in [1, 5]$ into pixelwise one-hot tensors. The relaxation effectively decomposes the problem of generating a target grain distribution to the easier problem of enforcing the volume fraction for each orientation.

The input latent vector, $\mathbf{z} \sim N(0, \mathbf{I})$ allows for exploration of the manifold of poly-crystalline microstructures, while $\mathbf{r} \in [0, 1]^5$ is a vector consisting of volume fractions of each of the five orientations. In order to ensure that the geometric constraint of adjoining grains being assigned different orientations while satisfying the statistical invariances; we use a similar technique as defined in sec. 4. The final layer of $G_\theta$ is a convolutional layer that outputs a tensor with the same number of channels as that of the orientations. We also use a softmax activation to ensure that each pixel is assigned to only a single orientation. We train InvNet with the alternating optimization technique with an additional statistical invariance applied channelwise;

$$L_I(\theta) = \mathbb{E}_{(\mathbf{z},\mathbf{r})} \| f_v(G(\mathbf{z}, \mathbf{r})) - \mathbf{r} \| \tag{9}$$

$$\text{where } f_v(\mathbf{x}) = \sum_{i=0}^{d} x_i / d \; ; \; \mathbf{x} \in \mathbb{R}^d$$

During inference, we use the *argmax* operation to convert the generated 3D tensors back to the original 2D polycyrstalline maps. Our experimental results in Fig. 8 show that InvNet successfully generate the microstructures satisfying the requisite grain distribution (volume fraction) accurately. We note that the invariance function(Eq. 10) only captures the relative volume fractions of each orientation. However, the grain shape and the appropriate placement of each grain are implicit features that InvNet learns from data. InvNet, therefore proves to be a useful technique to explore data spaces that have partially modelled dynamics.

Fig. 8 shows generated examples for different instances of required volume fractions. We observe that our model is able to generalize for a large class of grain distributions. We also analyse the performance of our model in reproducing images with a required target distribution. As evidenced by Fig. 7, InvNet is successful in generating polycrystalline microstructures with a required grain distribution for a large variety of cases. Interestingly, InvNet is able to generate images for distributions that are not found in the training dataset- for e.g. the highly skewed distributions in Fig. 7. Some examples of microstructure images generated for such cases have been presented in Fig. 8. Additional results showing generalization are also presented in the appendix (Fig. 15).

## 7 Discussion and Conclusion

We have proposed InvNets, a natural extension of GANs that enables specifying additional structural/statistical invariances in generated samples.

Our approach relies on representing the invariance as an additional loss term that we alternately optimize in addition to the standard adversarial training in GANs. We also show a number of stylized and real-world applications. Additionally, we present examples of InvNets enforcing constraints for integer valued solution spaces. Since InvNets also generalize to problems which have discrete solution spaces, we posit that our approach can be extended to a large class of ill-posed discrete optimization problems.

An important potential research direction would be to extend InvNets to 3D spaces for which GANs are currently infeasible. InvNets can also be extended to enforce other kinds of domain informed invariances; for e.g. PDEs governing the dynamics of a system. InvNets for structured domains such as graphs woudl further prove to be useful tools for more general problems.

Finally, the theoretical analysis of the equilibrium (for both limiting and non-limiting cases) and stability of training for the modified three-way game is an open question; especially for nonlinear discrete invariances.

## Acknowledgements

This work was supported by the DARPA AIE (AIRA) program under grant DARPA-PA-18-02-02, the National Science Foundation under grants NSF CCF-1750920 and NSF DMREF 1435587, U.S. AFOSR YIP grant FA9550-17-1-0220, GPU gift grants from the NVIDIA Corporation, and a faculty fellowship by the Black and Veatch Foundation. The authors would like to thank James Blackshire, Megna Shah, Daniel Sparkman, and Jiangying Zhou for inspiring discussions.

## References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *Proc. Int. Conf. Machine Learning (ICML)*, 214–223.

Blaschke, T.; Olivecrona, M.; Engkvist, O.; Bajorath, J.; and Chen, H. 2018. Application of generative autoencoder in de novo molecular design. *Molecular informatics* 37(1-2):1700123.

Brock, A.; Donahue, J.; and Simonyan, K. 2019. Large scale gan training for high fidelity natural image synthesis. In *Proc. Int. Conf. Learning Representations (ICLR)*.

Cahn, J. W., and Hilliard, J. E. 1958. Free energy of a nonuniform system. i. interfacial free energy. *The Journal of chemical physics* 28(2):258–267.

Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; and Abbeel, P. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Adv. Neural Inf. Proc. Sys. (NeurIPS)*.

Feng, J.; Teng, Q.; He, X.; and Wu, X. 2018. Accelerating multipoint statistics reconstruction method for porous media via deep learning. *Acta Materialia* 159:296–308.

Ganapathysubramanian, B., and Zabaras, N. 2007. Modeling diffusion in random heterogeneous media: Data-driven models, stochastic collocation and the variational multiscale method. *Journal of Computational Physics* 226(1):326–353.

Ganapathysubramanian, B., and Zabaras, N. 2008. A nonlinear dimension reduction methodology for generating data-driven stochastic input models. *Journal of Computational Physics* 227(13):6612–6637.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Adv. Neural Inf. Proc. Sys. (NeurIPS)*, 2672–2680.

Groeber, M. A., and Jackson, M. A. 2014. Dream.3d: A digital representation environment for the analysis of microstructure in 3d. *Integrating Materials and Manufacturing Innovation* 3:56–72.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. In *Adv. Neural Inf. Proc. Sys. (NeurIPS)*, 5767–5777.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *IEEE Conf. Comp. Vision and Pattern Recog*, 5967–5976.

Jiang, S.; Liu, H.; Wu, Y.; and Fu, Y. 2019. Spatially constrained generative adversarial networks for conditional image generation. *arXiv preprint arXiv:1905.02320*.

Jin, Y.; Zhang, J.; Li, M.; Tian, Y.; Zhu, H.; and Fang, Z. 2017. Towards the automatic anime characters creation with generative adversarial networks. *arxiv preprint arXiv:1708.05509* abs/1708.05509.

Korneev, S.; Narodytska, N.; Pulina, L.; Tacchella, A.; Bjorner, N.; and Sagiv, M. 2018. Constrained image generation using binarized neural networks with decision procedures. In Beyersdorff, O., and Wintersteiger, C. M., eds., *Theory and Applications of Satisfiability Testing (SAT)*, 438–449. Springer International Publishing.

Mirza, M., and Osindero, S. 2014. Conditional generative adversarial nets. *arxiv preprint, arXiv:1411.1784*.

Mokhtari, A.; Ozdaglar, A.; and Pattathil, S. 2019. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. *arXiv preprint arXiv:1901.08511*.

Odena, A.; Olah, C.; and Shlens, J. 2017. Conditional image synthesis with auxiliary classifier gans. In *Proc. Int. Conf. Machine Learning (ICML)*, 2642–2651.

Pokuri, B. S. S.; Shah, V.; Joshi, A.; Hegde, C.; Sarkar, S.; and Ganapathysubramanian, B. 2019. Binary 2d morphologies of polymer phase separation. *doi: 10.5281/zenodo.2580293*.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *arxiv preprint arXiv:1511.06434*.

Roberts, A. P. 1997. Statistical reconstruction of three-dimensional porous media from two-dimensional images. *Physical Review E* 56(3):3203.

Sanchez-Lengeling, B., and Aspuru-Guzik, A. 2018. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* 361(6400):360–365.

Stinis, P.; Hagge, T.; Tartakovsky, A. M.; and Yeung, E. 2018. Enforcing constraints for interpolation and extrapolation in generative adversarial networks. *arxiv preprint, arxiv:1803.08182*.

Tahmasebi, P.; Hezarkhani, A.; and Sahimi, M. 2012. Multiple-point geostatistical modeling based on the cross-correlation functions. *Computational Geosciences* 16(3):779–797.

Torquato, S. 2013. *Random heterogeneous materials: microstructure and macroscopic properties*, volume 16. Springer Science & Business Media.

Wodo, O., and Ganapathysubramanian, B. 2012. Modeling morphology evolution during solvent-based fabrication of organic solar cells. *Computational Materials Science* 55:113–126.

Wu, J.; Zhang, C.; Xue, T.; Freeman, B.; and Tenenbaum, J. 2016. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Adv. Neural Inf. Proc. Sys. (NeurIPS)*.

Yeong, C. L. Y., and Torquato, S. 1998. Reconstructing random media. *Phys. Rev. E* 57:495–506.

Zhao, S.; Ren, H.; Yuan, A.; Song, J.; Goodman, N. D.; and Ermon, S. 2018. Bias and generalization in deep generative models: An empirical study. In *Adv. Neural Inf. Proc. Sys. (NeurIPS)*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Proc. IEEE Intl. Conf. Comp. Vision*.

# Appendix

***Notation.*** We represent vectors in lowercase boldface, $\mathbf{v}$ whereas matrices are uppercase boldface, $\mathbf{V}$. Given an image, $\mathbf{U}$, we represent the spatial derivative with respect to axis, $x$ as $\mathbf{U}_x$. Additionally, the gradient of a scalar, $L$ with respect to a vector, $\mathbf{a}$ is represented as $\nabla_a L$. $\mathbf{I}$ represents identity matrix.

## A Detailed Related Work

**Generative Adversarial Networks.** GANs (Goodfellow et al. 2014) are a popular approach for modelling real world data distributions. The standard adversarial training approach involves optimizing a mini-max game between a generator and a discriminator defined by an approximate Jensen-Shannon divergence. This necessarily leads to unstable training and requires careful tuning of relevant hyperparameters. Mao *et al.*solve the problem of vanishing gradients by optimizing the discriminator with least squares instead of cross-entropy. Arjovsky *et al.* (Arjovsky, Chintala, and Bottou 2017) and Gulrajani *et al.* (Gulrajani et al. 2017) propose a more stable version of GAN by modifying the discriminator loss to estimate the Wasserstein-1 distance between the data distribution and the generator output. A major disadvantage for such models is that the inability to control the generator output in any way. Zhao *et al.* (Zhao et al. 2018) study the generalization in GANs.

Conditional GANs (Mirza and Osindero 2014) provide a solution by conditioning the generator on categorical labels so as to control the class of outputs that are generated. Chen *et al.* (Chen et al. 2016) extend this to allow control over specific semantic parameters such as stroke width in the case of handwritten digits.

Such generative models therefore offer a mechanism to accurately represent complex data spaces without knowing the entire topology. Consequently, GANs are often used as representations of solution spaces to complex physical and dynamical systems.

**Invariance in generative models.** The problem of training a generative model to generate samples from a specific distribution is often solved through the use of data. This approach is extremely useful when the data can not be modeled mathematically. However, for many applications, there exist at least partial mathematical definitions for training data. These mathematical definitions act in place of data, acting as constraints to define the support of the generator distribution.

Stinis *et al.* (Stinis et al. 2018) employ a noisy data training approach with mathematical constraints in order to interpolate and extrapolate on the generator distribution. Contrary to their approach of weakening the discriminator training with noisy inputs, we use an alternating minimization scheme to force the discriminator to respect the invariances. Jiang *et al.*(Jiang et al. 2019) use segmentation masks as constraints to enforce structural conditions to generate face images. Svyatoslav (Korneev et al. 2018), on the other hand, enforce a PDE as a constraint by using a binary neural network as a PDE solver using decision processes. Their approach is restricted to the special case of generating binary images, whereas our algorithm is more general and can enforce any continuous and differentiable invariance.

**Microstructure generation.** An entire sub-field in computational material science is devoted to the development of methods for the *simulation* of microstructures (Ganapathysubramanian and Zabaras 2008; 2007; Roberts 1997) and subsequent quantification (Ganapathysubramanian and Zabaras 2008; 2007). Here, microstructure realizations are synthesized that satisfy certain target statistical properties of the material distribution. Several strategies were developed for microstructure generation using both analytical approaches and optimization approaches. Examples of such methods include Gaussian random fields (Roberts 1997), optimization-based methods (Yeong and Torquato 1998), multi-point statistics (Feng et al. 2018), and layer-by-layer reconstruction (Tahmasebi, Hezarkhani, and Sahimi 2012).These statistical properties could be scalars (such like total volume fraction of a material) or more complex functions (like 2-point correlations and other material statistics) (Torquato 2013). Recent advances also involve the generative modeling techniques (Sanchez-Lengeling and Aspuru-Guzik 2018), however, those largely rely on the availability of training datasets.

For our experiments in generating microstructures, we use the Binary 2D microstructures dataset (Pokuri et al. 2019) based on Cahn-Hilliard equation (Cahn and Hilliard 1958) for training and testing.

## B Microstructures Generation using InvNet: Additional Details

### Motivation

An overarching theme of materials research is the design of material distributions (also called microstructure) so that the ensuing material exhibits tailored properties. In microstructure-sensitive design, quantifying the effect of microstructure features on performance is critical for the efficient design of application-tailored devices. Microstructures are represented as binary images indicating the arrangement of constituent materials within the mixture. The statistical properties of such microstructural images are useful in predicting the physical and chemical properties of the mixture material - thus aiding into faster material discovery. To obtain a material with desired property, a microstructure having the corresponding statistical property need to be generated. We feed in such statistical properties as the invariances in our framework, and come up with a generative model that can sample from the set of all microstructures adhering to desired statistical properties. We propose both a data-driven and data-free generative network for synthetic microstructures adhering the invariances for the training.

### Preliminaries

In the context of microstructure generation problem, we consider the underlying material to be a two-phase homogeneous, isotropic material. Our setup for statistical characterization of microstructure follows with Torquato *et al.* (Torquato 2013). Consider an instance of the two-phase

homogeneous isotropic material within $d$-dimensional Euclidean space $\mathbb{R}^d$ (where $d \in \{2, 3\}$). A phase function $\phi(\cdot)$ is used to characterize this two-phase system, defined as:

$$\phi^{(1)}(\mathbf{r}) = \begin{cases} 1, \mathbf{r} \in V_1, \\ 0, \mathbf{r} \in V_2, \end{cases} \tag{10}$$

where $V_1 \in \mathbb{R}^d$ is the region occupied by phase 1 and $V_2 \in \mathbb{R}^d$ is the region occupied by phase 2.

Given this microstructure defined by the phase function, $\phi$, statistical characteristics can be evaluated. These include the $n-$moments, ($n$-point correlation functions) for $n = 1, 2, 3, ....$ For homogeneous and isotropic media, These depend neither on the absolute positions of $n-$points, nor on the rotation of these spatial co-ordinates; instead, they depend only on relative displacements. The $1^{st}$-moment, $p_1$, commonly known as *volume fraction*, is constant throughout the material. The volume fraction of phase 1, $p_1^{(1)}$, is defined as:

$$p_1^{(1)} = \mathbb{E}_{\mathbf{r}} \phi^{(1)}(\mathbf{r}).$$

The $2^{nd}-$moment is a function of $r$ and is defined as:

$$\mathbf{p}_2^{(1)}(r_{12}) = \mathbb{E}_{\mathbf{r_1}, \mathbf{r_2}} \left[ \phi^{(1)}(\mathbf{r}_1) \phi^{(1)}(\mathbf{r}_2) \right].$$

The $2^{nd}$ moment (known as $2-$point correlation as well) is one of the most important statistical descriptors of microstructures. An alternate interpretation of $2^{nd}$ moment is the probability that two randomly chosen points $\mathbf{r}_1$ and $\mathbf{r}_2$ a certain distance apart both share the same phase.

Henceforth we omit the superscript representing the phase and subscripts representing the spatial points for simplicity, and refer to volume fraction as $p_1$, and 2-point correlation as $\mathbf{p}_2$. It can be shown that $\mathbf{p}_2(r = 0) = p_1$ and $\lim_{r \to \infty} \mathbf{p}_2(r) = p_1^2$.

In the training step, we use the above statistical properties as invariances for training the InvNet. The invariance loss $L_I(\cdot)$ can be defined as $l_2-$ loss:

$$L_I = \lambda_1 \| f_{p_1}(G_\theta(\mathbf{z}, \mathbf{p}_2^*)) - p_1^* \|_2^2 + \lambda_2 \| f_{\mathbf{p}_2}(G_\theta(\mathbf{z}, \mathbf{p}_2^*)) - \mathbf{p}_2^* \|_2^2 \tag{11}$$

where $f_{p_i}$ represent the functional forms of the moments; $p_1^*, \mathbf{p}_2^*$ are target values of the moments. The coefficients, $\lambda_i$ are appropriately chosen for the tasks to be solved.

**Dataset.** We use the Binary 2D microstructures dataset (Pokuri et al. 2019) based on Cahn-Hilliard equation (Cahn and Hilliard 1958) for training and testing. The dataset contains $\sim 34k$ binary microstructures of size $101 \times 101$ obtained by sampling the evolving solutions across time. The dataset contains images with diverse values of $1^{st}$ and $2^{nd}$ moments, and implicitly exhibit higher moments too. For training, we resize the images to $64 \times 64$.

## C Additional Results.

**Comparison with Stinis et al..** We use ideas from Stinis et al. (ECF) and modify a simple WGAN-Gp to enforce the $1^{st}$ moment constraint. This involves two steps: (1) Modify the discriminator architecture to take as input the data and the corresponding residual of the $p_1$ invariance loss, (2) Add noise to the real data residual to ensure stability. Since teh authors have not shared the code, we implement the approach for comparison.

We observe that for our application; the ECF algorithm fails to converge. In fact, the discriminator and the generator loss explode to very high values. We hypothesize that this may be a consequence of the model failing to model the joint distribution due to large residual values in teh intiial steps.

**Generating shapes with conditional GANs.** For comparison, we train a cGAN with our dataset of circles. As a modification of the original vanilla cGAN, we feed the centroids and radii of both circles from real data sample instead of the labels. Additionally, we train with the WGAN-GP objective for smoother training.

The generator takes the same input as the InvNet by $[\mathbf{z}, \mathbf{r}]^T$ while the discrimator is fed the generated or true images; stacked with $\mathbf{r}$; a vector consisting of the target centroids and radii. We observe that cGAN completely fails to learn the distribution of the dataset as shown in Fig 9. Additionally, Fig. 1 shows that cGAN fails to generate images obeying the target invariance and in fact learns the training data distribution instead.

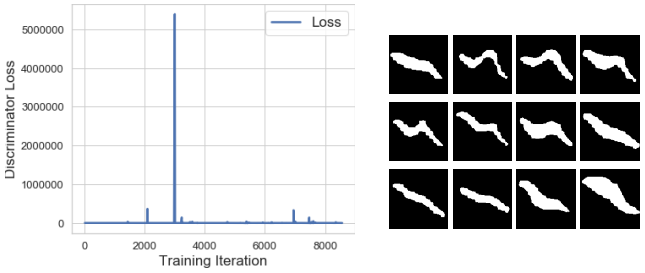



Figure 9: *Training results from cGAN. The discriminator loss shows the failure of the model for the toy dataset. Apart from the geometrical invariance, we also observe from generated images that cGAN model collapse t with two circles dataset.*

**Comparison with AC-GANs.** Similar to the experiment above, we train an AC-GAN for the same task. For fair comparison, we use the WGAN objective and the softmax generator with equal number of parameters as InvNet. Additionally, as per the original AC-GAN paper (Odena, Olah, and Shlens 2017), we modify the discriminator architecture to take in the true or generated data along with the encoded target vector. The AC-GAN is trained with ADAM (learning rate=$0.001$) for $\sim 23k$.

While the AC-GAN successfully learns the input data distribution, it fails to capture the association between the input target properties and those of the generated images. Additionally, the convergence of the AC-GAN model is far slower than that of InvNet.

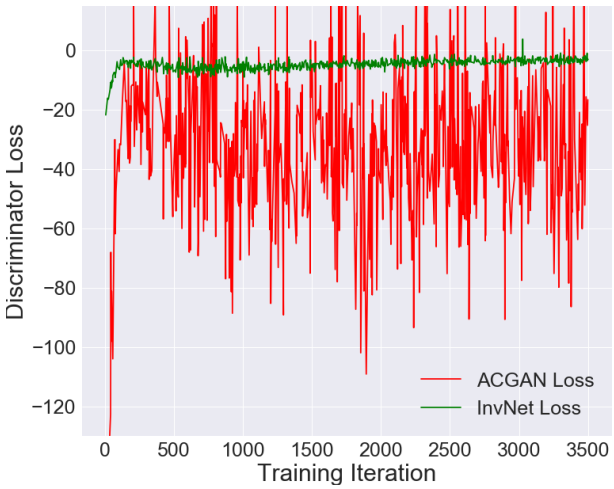



Figure 10: *Comparison of the discriminator loss curves for ACGAN and InvNet. Note the smoother training curve for InvNet.*

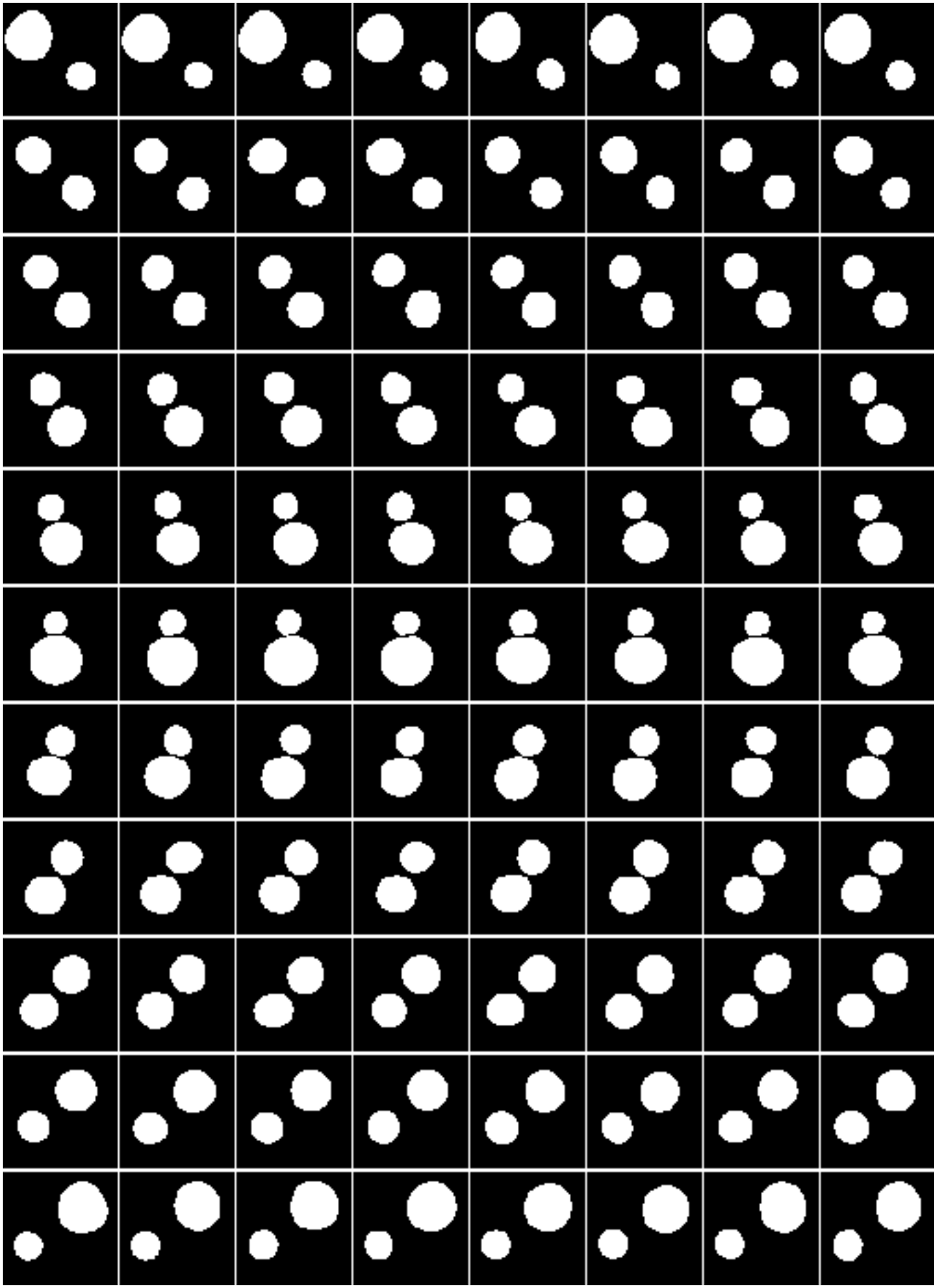


Figure 11: *Additional results for the toy dataset with two circles. Each row corresponds to the specific geometrical constraints.*

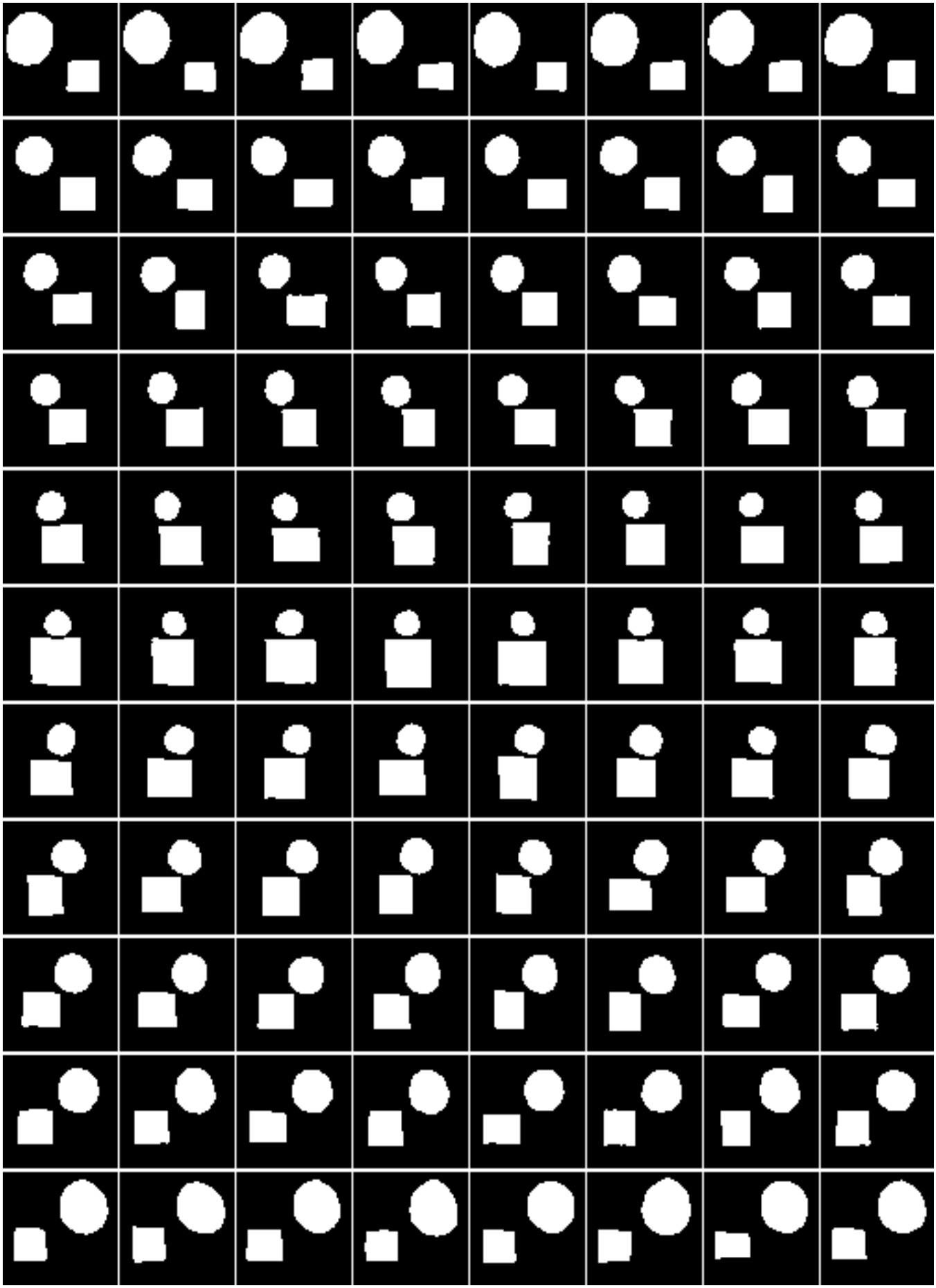


Figure 12: *Additional results for the toy dataset with a circle and a square. Each row corresponds to the specific geometrical constraints.*

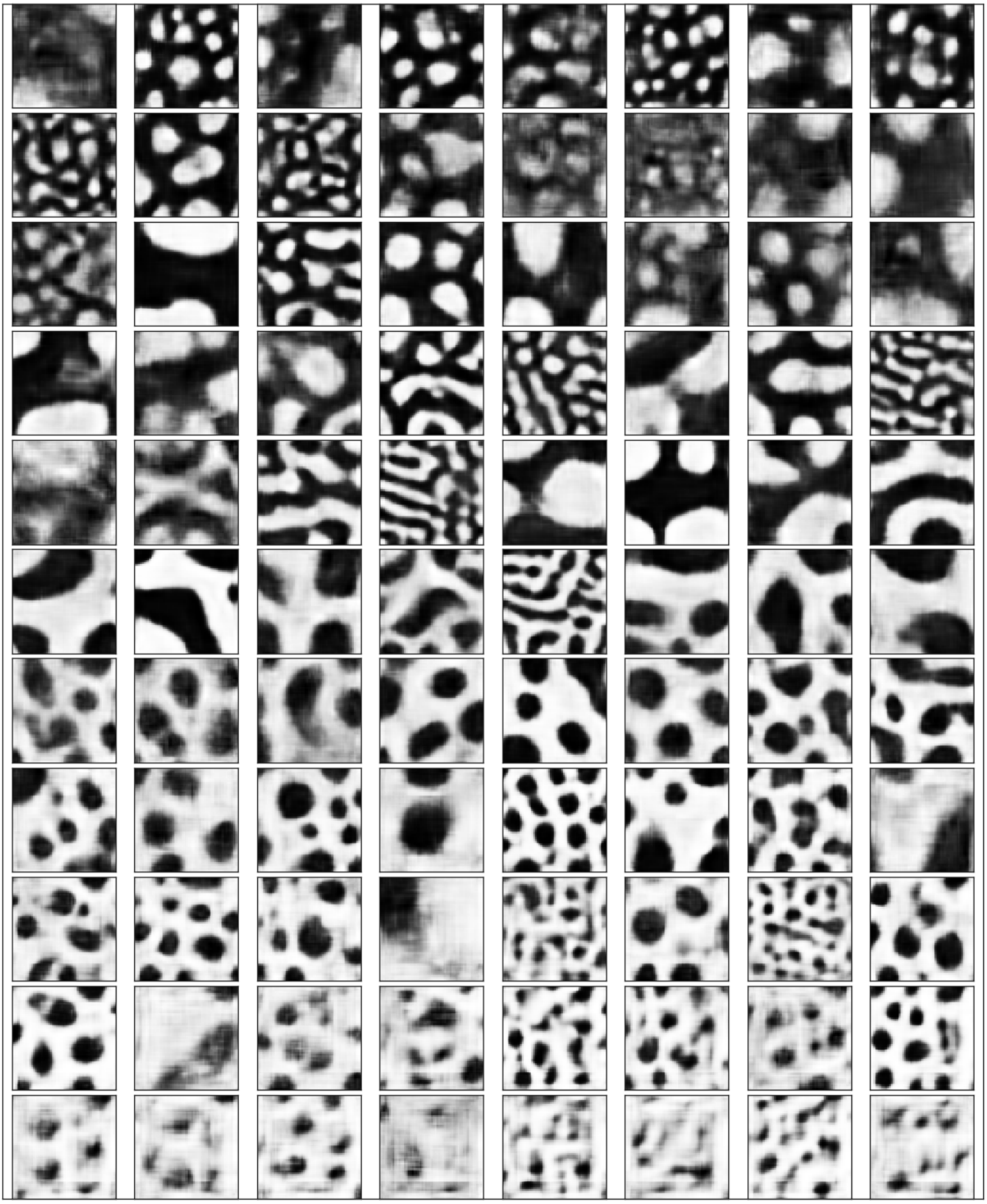


Figure 13: Additional results for binary microstructure generation using the volume fraction control parameter. Each row corresponds to a specific volume fraction ($p_1$) value ranging from $0.3$ to $0.8$.

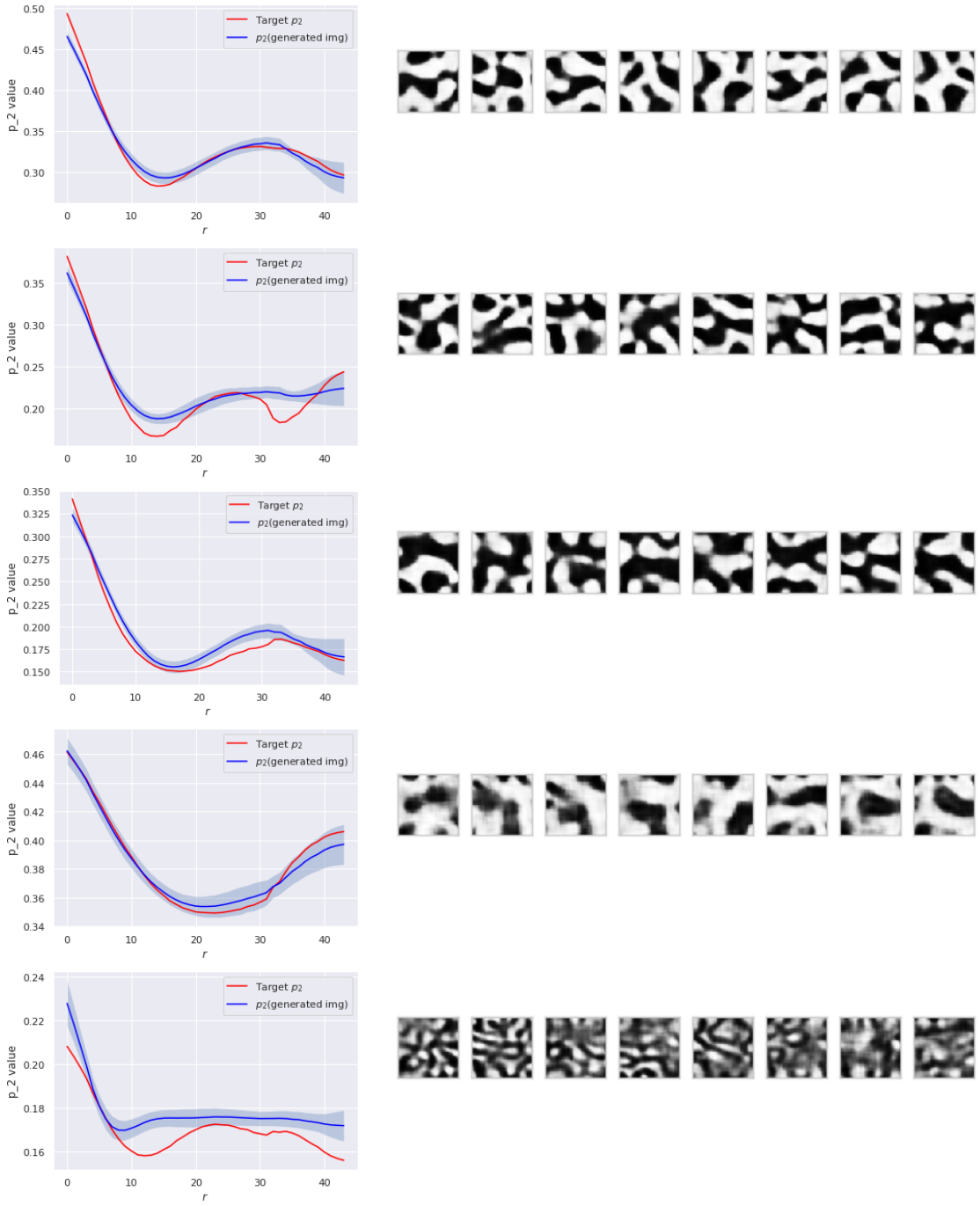



Figure 14: *Results for InvNet trained to generate binary microstructures given target* $\mathbf{p}_2$ *curves. The plot displayes the target and the mean curve for* 64 *generated images. The corresponding images have been generated for the target* $\mathbf{p}_2$ *on the left for varying latent random latent vector,* $\mathbf{z}$*.*

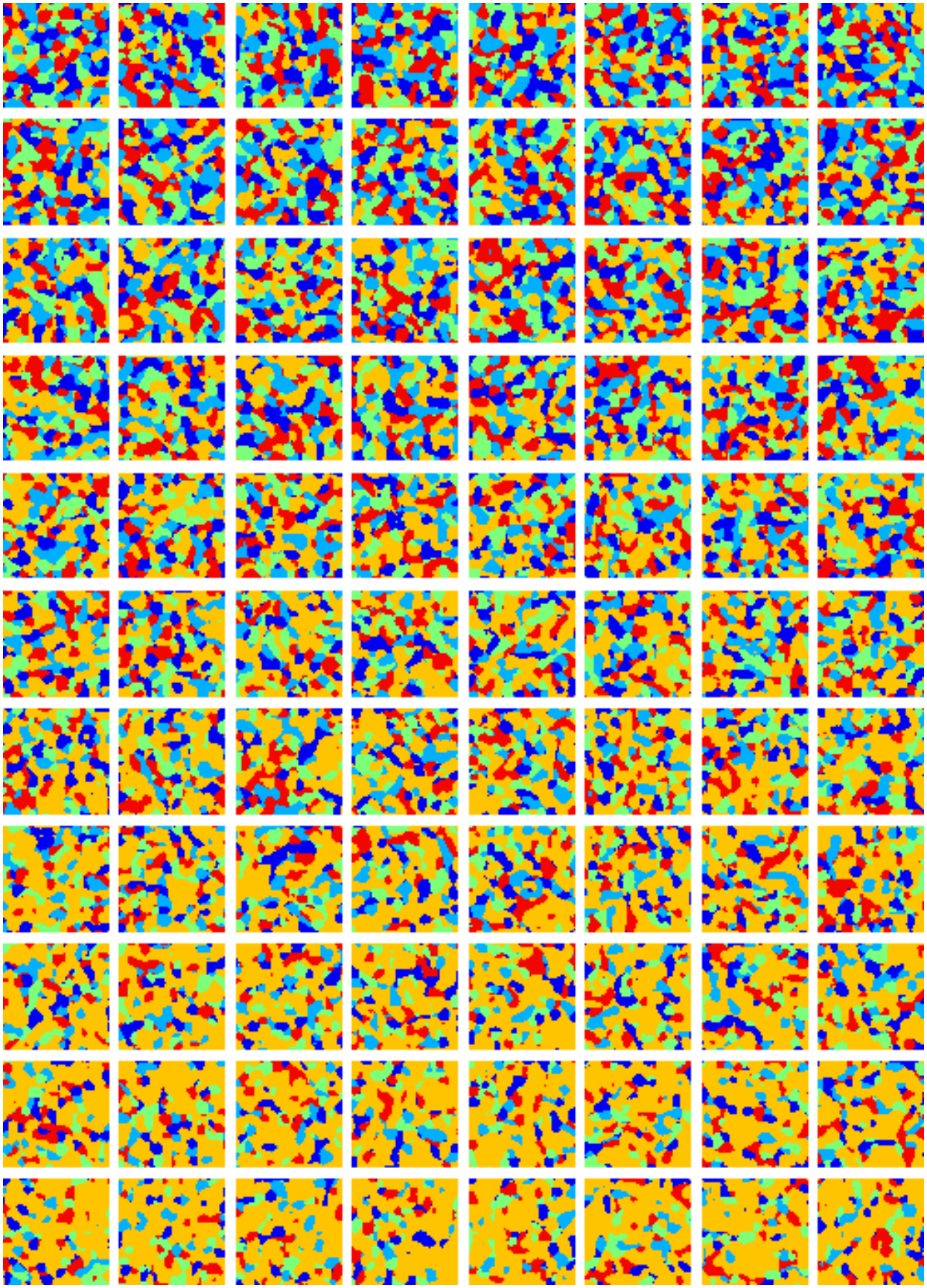


Figure 15: *Additional results for polycrystalline microstructure generation using the volume fraction control parameter. Each row corresponds to a specific vol. frac. value of a fourth orientation ranging from* $0.15$ *to* $0.65$*. Note that the range of the fourth orientation from the dataset is in* $0.18$ *to* $0.50$*. Last three rows (with volume fraction* $(p_1)$ *corresponding to* $0.55, 0.60, 0.65$*) generated from the interpolated distribution from its data distribution.*

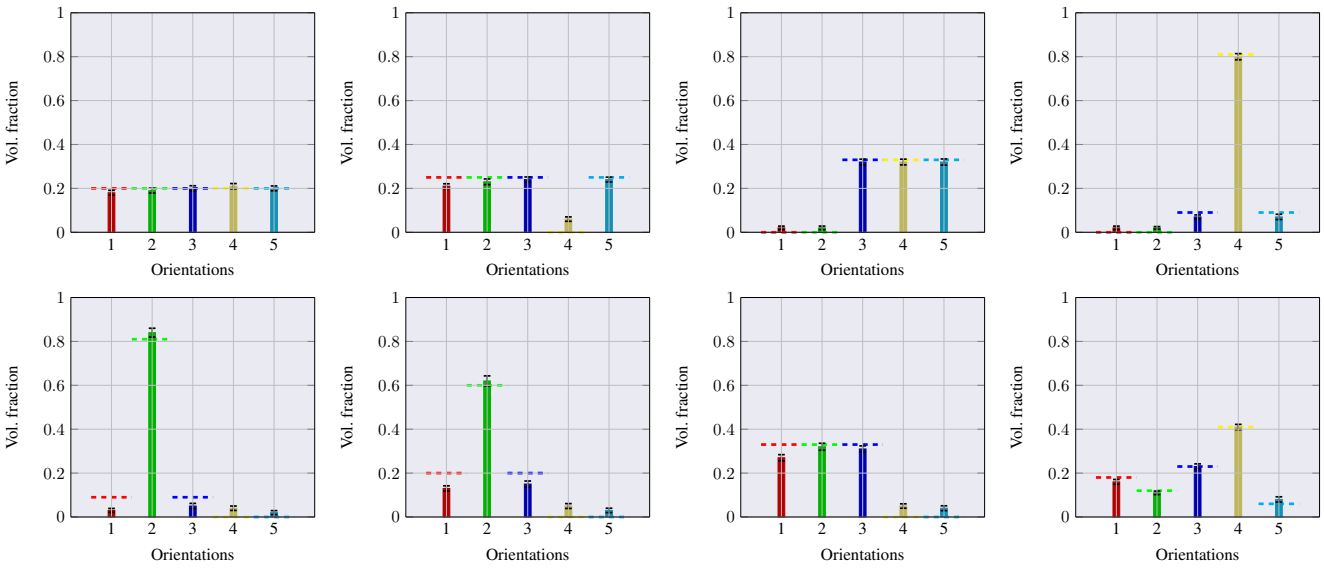



Figure 16: *Grain distribution for generated polycrystalline data. The plots represent the mean volume fractions of* 100 *images for each required distribution. The horizontal* dashed *lines denote the target volume fraction for the orientation. Observe that InvNet is successful at generating polycrystalline images for a large variety of grain distributions. Also note that several of the target distributions are not present in the training dataset.*

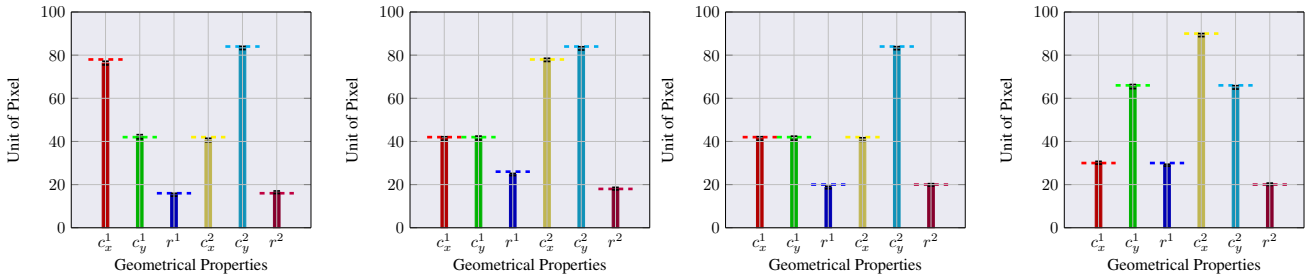



Figure 17: *Performance of InvNet in respecting target invariances. The bar plots show the centroids* $(c_x^i, c_y^i)$ *and radii,*$r^i$ *for images generated by InvNet for the* two circle *task. Note that InvNet successfully generates images that respect the required target invariance value (represented by the dashed horizontal line) with very low error.*