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, ϕ, 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, λ_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 101×101 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 64×64.

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.

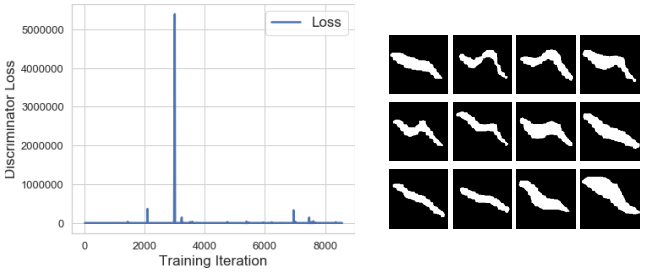

Figure 9: *Training results from cGAN. The discriminator loss shows the failure of the model for the toy dataset. Apart from the geometrical invariance, we also observe from generated images that cGAN model collapse t with two circles dataset.*

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.

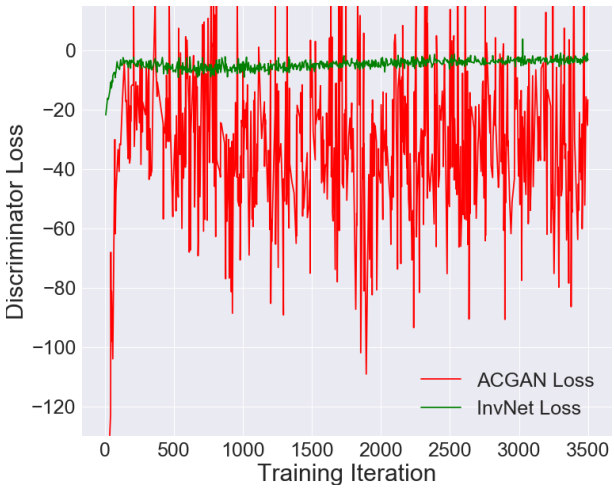

Figure 10: *Comparison of the discriminator loss curves for ACGAN and InvNet. Note the smoother training curve for InvNet.*

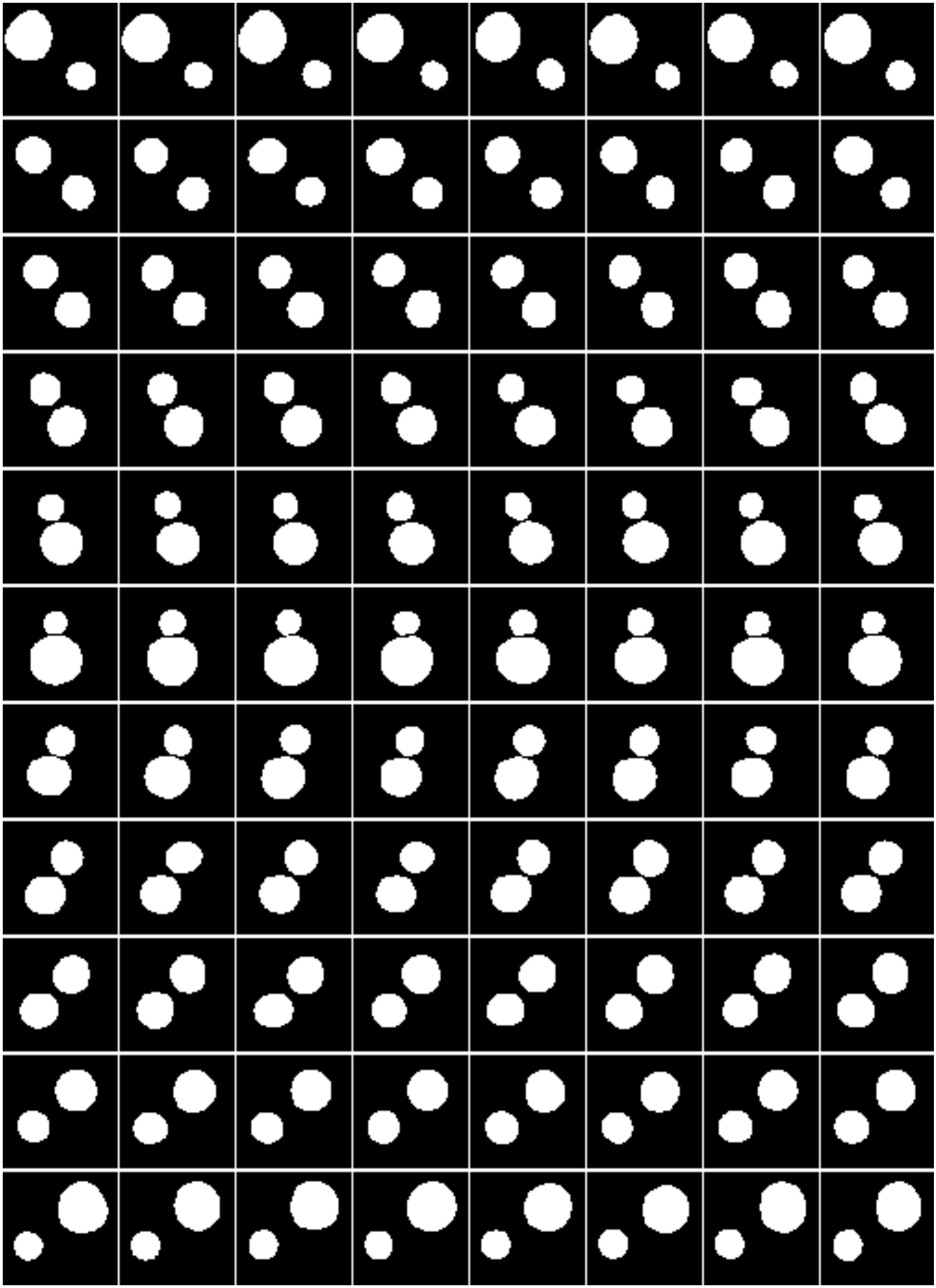

Figure 11: *Additional results for the toy dataset with two circles. Each row corresponds to the specific geometrical constraints.*

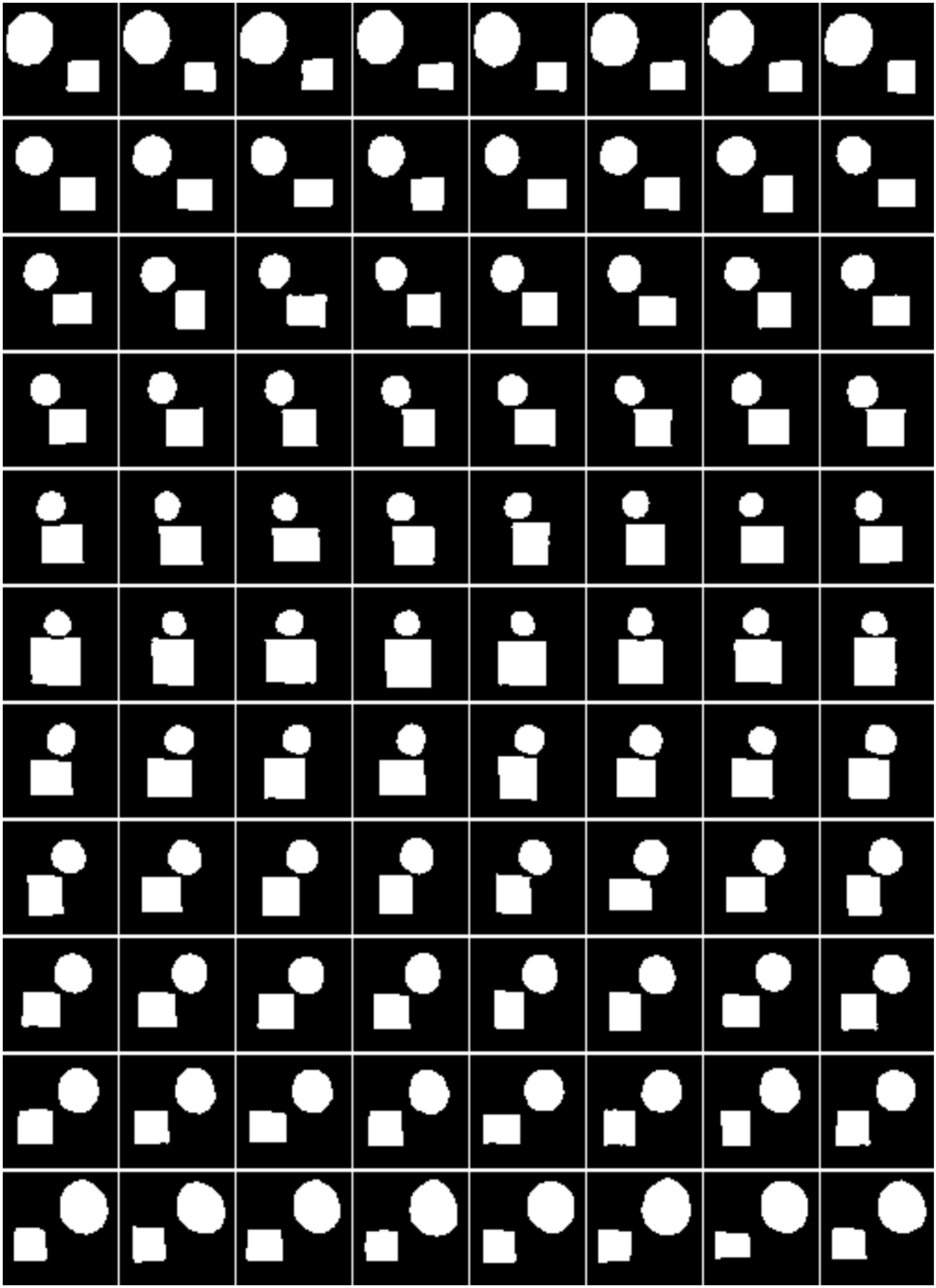

Figure 12: *Additional results for the toy dataset with a circle and a square. Each row corresponds to the specific geometrical constraints.*

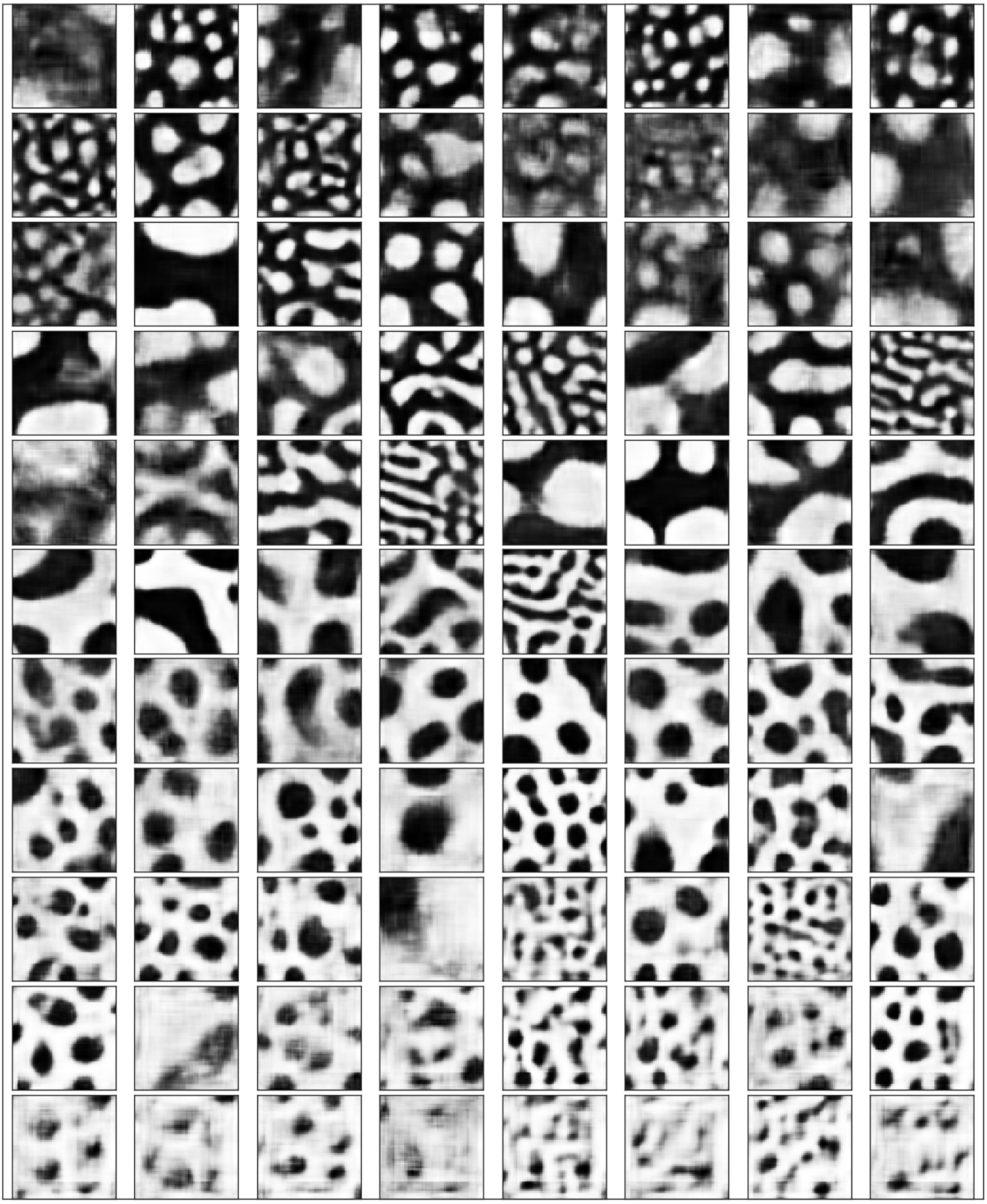

Figure 13: Additional results for binary microstructure generation using the volume fraction control parameter. Each row corresponds to a specific volume fraction (p_1) value ranging from 0.3 to 0.8.

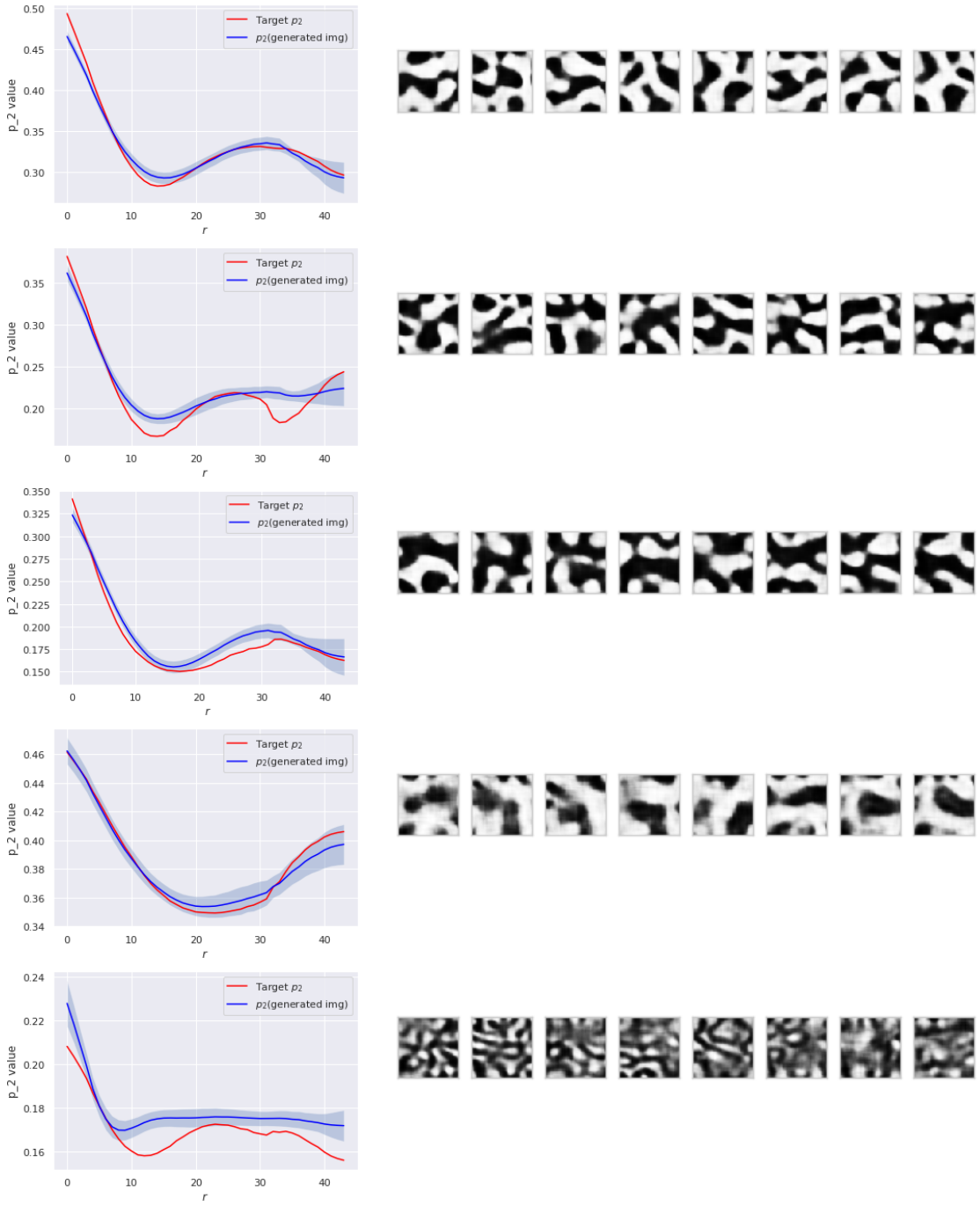

Figure 14: *Results for InvNet trained to generate binary microstructures given target* $\mathbf{p}_2$ *curves. The plot displayes the target and the mean curve for* 64 *generated images. The corresponding images have been generated for the target* $\mathbf{p}_2$ *on the left for varying latent random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

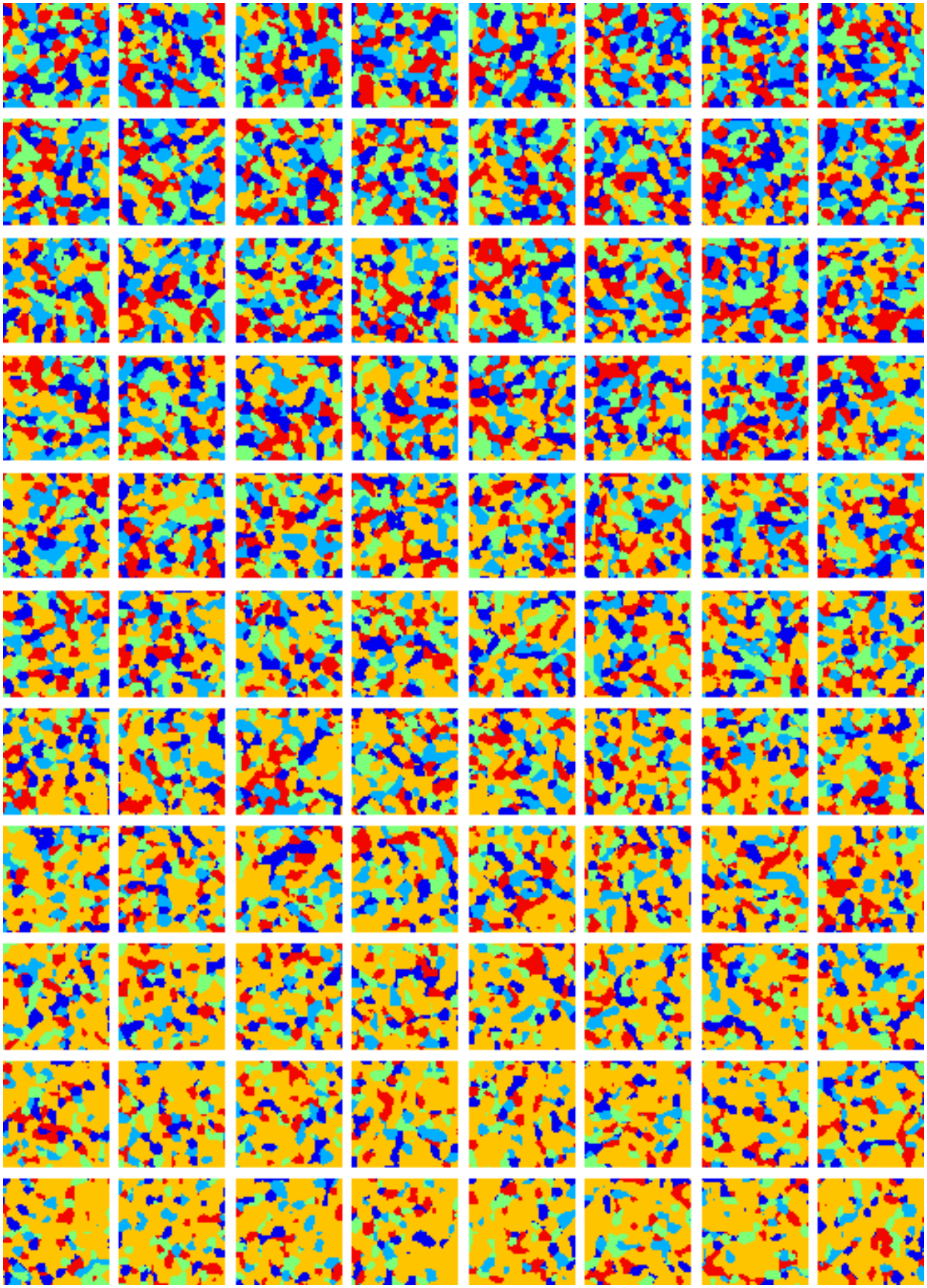

Figure 15: *Additional results for polycrystalline microstructure generation using the volume fraction control parameter. Each row corresponds to a specific vol. frac. value of a fourth orientation ranging from* 0.15 *to* 0.65*. Note that the range of the fourth orientation from the dataset is in* 0.18 *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.*

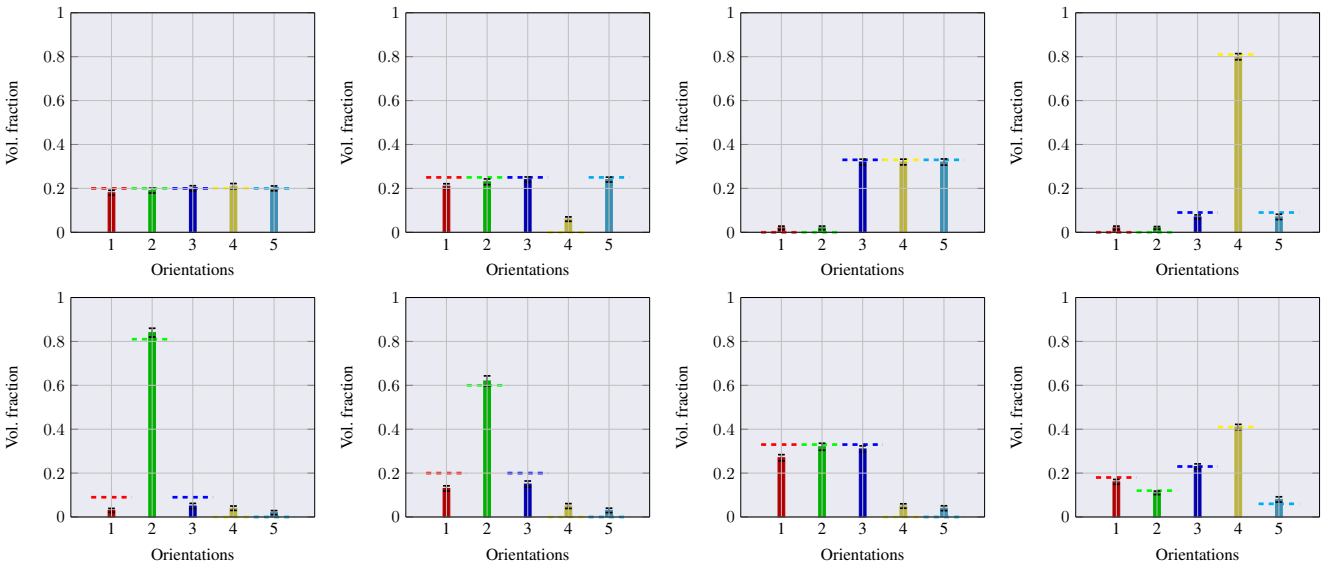

Figure 16: *Grain distribution for generated polycrystalline data. The plots represent the mean volume fractions of* 100 *images for each required distribution. The horizontal* dashed *lines denote the target volume fraction for the orientation. Observe that InvNet is successful at generating polycrystalline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

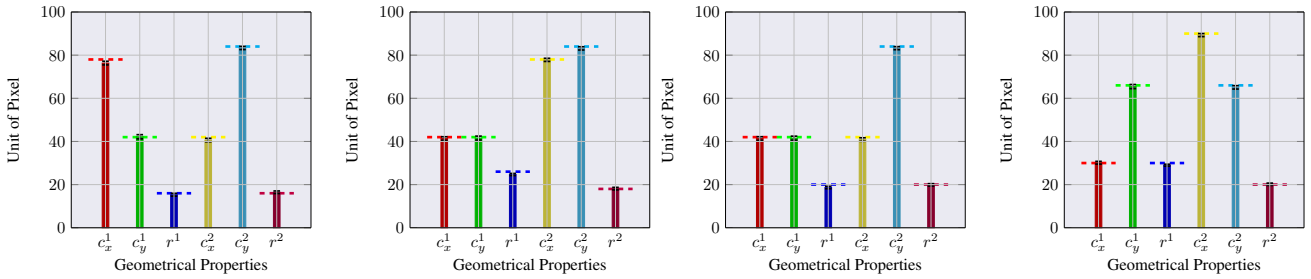

Figure 17: *Performance of InvNet in respecting target invariances. The bar plots show the centroids* (c_x^i, c_y^i) *and radii,*r^i *for images generated by InvNet for the* two circle *task. Note that InvNet successfully generates images that respect the required target invariance value (represented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