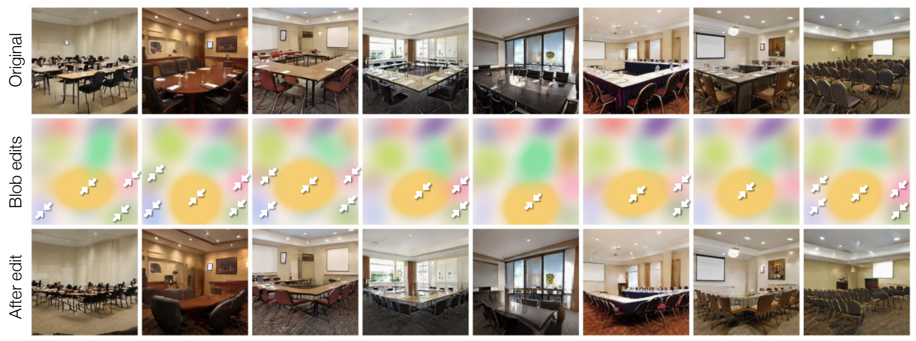


Fig. 13. Honey, I shrunk the conference room! As in Figure 3, we show the effect of resizing blobs in generated images. Here, we resize blobs corresponding to tables and chairs, and render identical rooms with shrunken furniture.

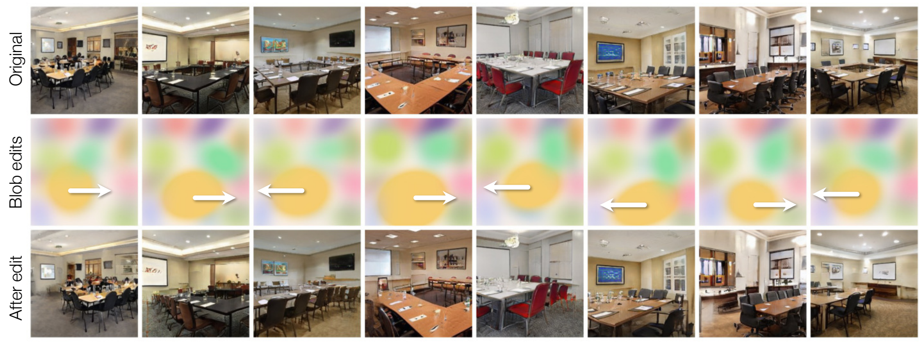


Fig. 14. Moving desks and chairs (conference room): As in Figure 4, we show the effect of moving blobs in generated images. Here, we move blobs corresponding to tables and chairs, and render identical rooms with shifted furniture.

Appendix A BlobGAN on other datasets

In the main text, we primarily showed results on LSUN bedrooms [97]. Below, we show that our model can be applied to other datasets and room types. We provide qualitative and quantitative results on our models trained on the challenging LSUN conference room dataset, as well as a joint dataset combining LSUN kitchens, dining rooms, and living rooms [97]. As with bedrooms, our model's images are competitive with previous work in terms of photorealism, and in addition allow extensive manipulation of images. Please see Table 3 for quantitative evaluation. We show image samples and edits on them in Figures 19, 13, 14, 15, 23, 22, and 16.

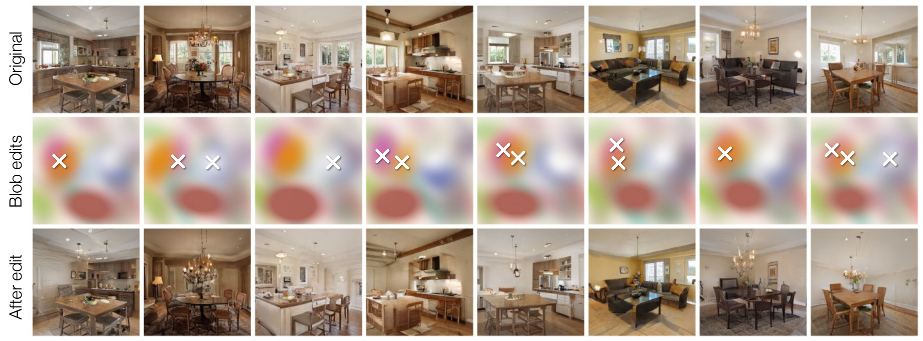


Fig. 15. Removing some or all windows (kitchen, living room, dining room): As shown in Figure 5, we can remove windows from complex scenes, though they are often hidden behind cluttered configurations of furniture. We can control which windows to remove by selecting only some of the relevant blobs.

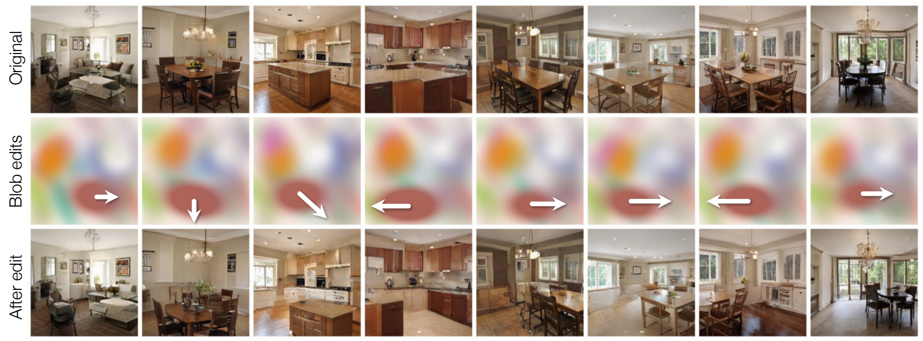


Fig. 16. Moving tables and chairs (kitchen, living room, dining room): Our representation can easily move tables and any associated chairs, by changing the location of blobs 42 (table) and 30 (chairs). Since the two move together, we only show one arrow to represent the edit.

Appendix B Modeling real images with BlobGAN

We show additional results on inversion and editing of real images in Figures 17 and 18. Images are drawn from the LSUN bedrooms validation set, which our model does not see during the training process.

B.1 Implementation details

In Section 4.5 and Figure 12, we demonstrate that real images can be inverted and manipulated with our model. Here, we provide additional details regarding the encoder training procedure. We take an encoder architecture E in the same form as the StyleGAN2 [39] discriminator, without mini-batch statistic discrimination. We use E for inverting images by having the last layer output a long flat vector, which we segment into blob parameters. In addition to reconstructing both real and synthetically generated images with LPIPS [103] and L2 penalties, we require the parameters $\hat{\boldsymbol{\beta}}$ to match the ground truth parameters $\boldsymbol{\beta}$ in

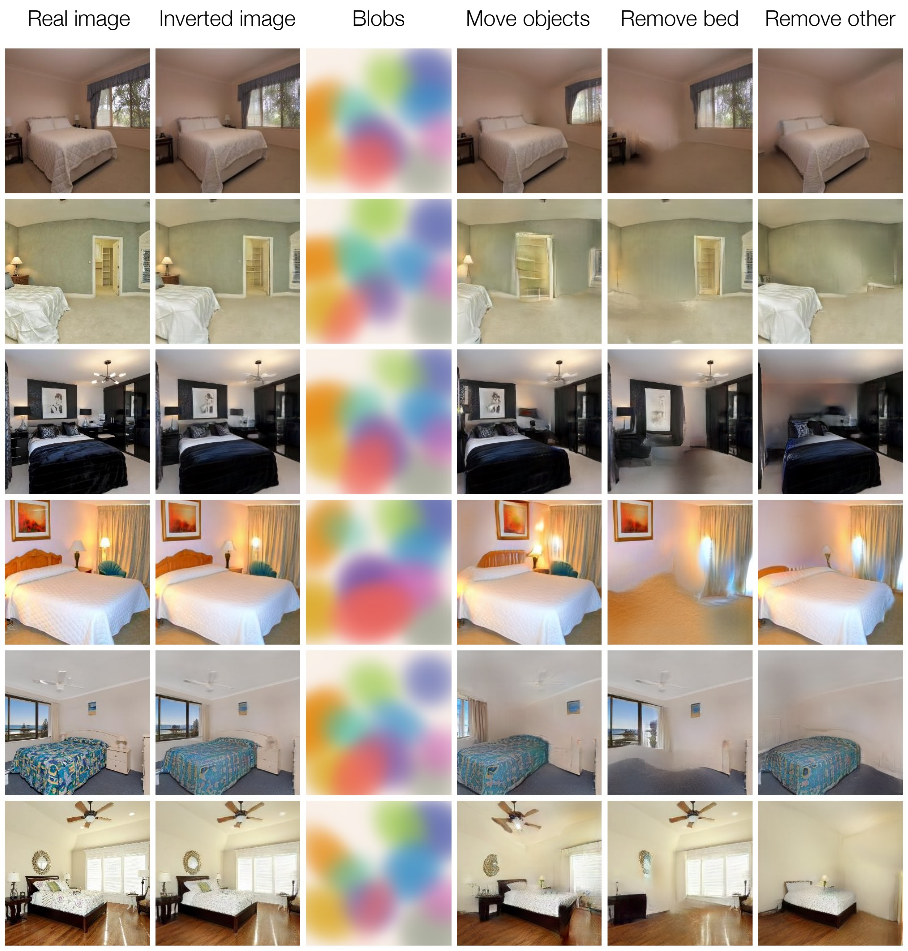


Fig. 17. Parsing real images via inversion: We show the flexibility of our learned representation by applying edits to real images inverted into blob space. We can remove and reposition objects in real images – spot the differences from the original!

the case of inverting generated images. Our overall loss is:

$$\begin{aligned}\mathcal{L}_{\text{inversion}} = &\mathcal{L}_{\text{LPIPS}}\left(\mathbf{x}_{\text{real}}, G(E(\mathbf{x}_{\text{real}}))\right) + \mathcal{L}_{\text{LPIPS}}(\mathbf{x}_{\text{fake}}, G(E(\mathbf{x}_{\text{fake}}))) \\ &+ \mathcal{L}_2(\mathbf{x}_{\text{real}}, G(E(\mathbf{x}_{\text{real}}))) + \mathcal{L}_2(\mathbf{x}_{\text{fake}}, G(E(\mathbf{x}_{\text{fake}}))) \\ &+ \lambda\mathcal{L}_2(\boldsymbol{\beta}_{\text{fake}}, E(\mathbf{x}_{\text{fake}})),\end{aligned} \tag{5}$$

with $\lambda = 10$ controlling the strength of the blob reconstruction loss. Taking the L2 loss on blob parameters as a flattened vector would heavily emphasize reconstructing the high-dimensional features, over the important, low-dimensional, scalar quantities of blob locations and sizes. Instead, we compute L2 separately over each blob attribute and take the mean.

We then further optimize the blob parameters to reconstruct the target image, with LPIPS and L2 losses and the Adam optimizer [40] with learning rate

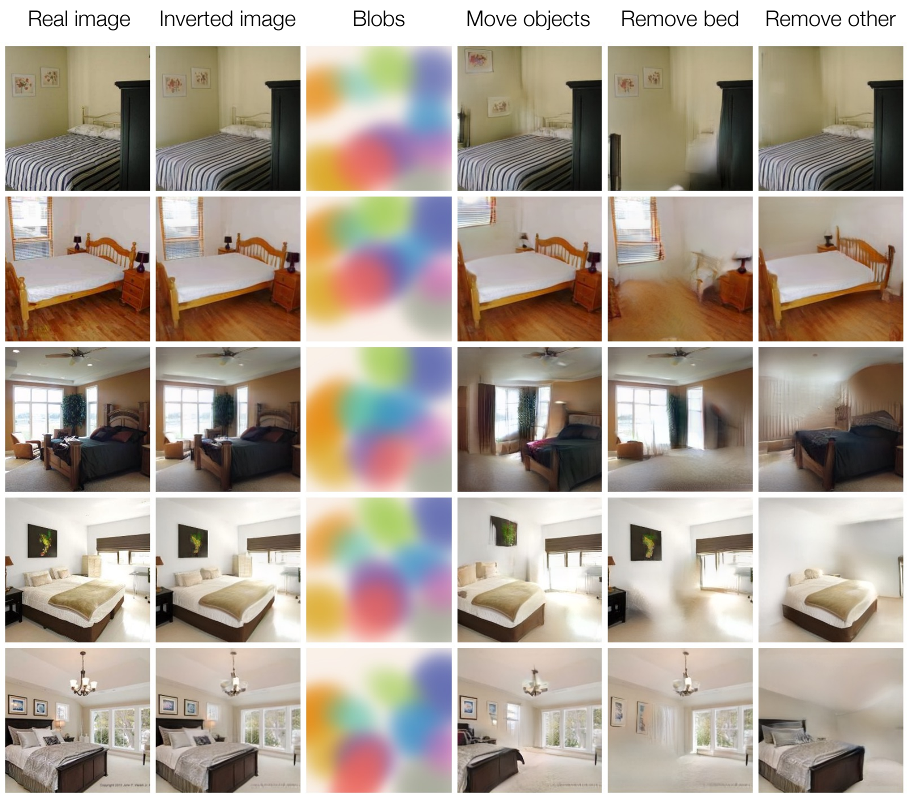


Fig. 18. Parsing real images via inversion: More results on inversion of real images.

0.01 for 200 steps. While better fitting the input image, this method potentially deviates from the manifold of latents that yield realistic images [1, 70], thus severely impeding editing abilities. Previously proposed solutions offer regularizations to keep the latents on this "manifold" [92, 105, 93]. However, we find our blob representation to be more robust in this sense, and latents yielded by this naïve optimization still amenable to editing.

Appendix C Blob parametrization

We represent the blob aspect ratio a as two scalar outputs a_0, a_1, sigmoided and then normalized to have a fixed product $a_0 a_1$; we find this to train more stably than one aspect ratio. We represent the blob angle θ with two scalars e_0, e_1, from which we construct a unit-normalized axis of rotation $\mathbf{e}$. We find this representation to train far more stably than others, such as regressing to a scalar θ or other parametrizations of Σ like log-Cholesky [48, 76].

We also experimented with alternate representations, such as closed-form ellipses and rectangles as well as Gaussian mixture models. However, we found gradient flow to blob parameters ill-behaved with rectangles and other explicitly

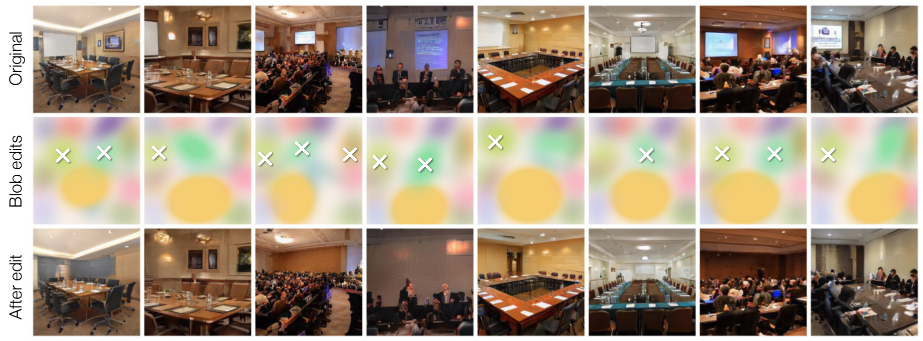


Fig. 19. Removing screens in conference rooms. As in Figure 5, we show the effect of removing certain blobs from generated images. Here, we remove blobs corresponding to screens from images of conference rooms.

	LSUN Conference			LSUN Kitchen+Living+Dining		
	FID ↓	Precision ↑	Recall ↑	FID ↓	Precision ↑	Recall ↑
StyleGAN2 [39]	6.21	0.5475	0.4554	4.63	0.6005	0.4397
BlobGAN	6.94	0.5297	0.4485	4.41	0.5818	0.4661

Table 3. On challenging collections of conference rooms and various types of indoor rooms in homes, our model is highly competitive with a StyleGAN2 baseline, while enabling all the applications of the BlobGAN representation. Our model outperforms StyleGAN2 given an equal number of gradient steps (1.5M) on the difficult union of various LSUN indoor scene categories, as measured by FID.

defined shapes, even with tricks like gradual opacity falloff, and these models failed to train. With GMMs, depth ordering and occlusions are lost, and blob size and shape depend on other blobs, harming performance. Our model is robust w.r.t. c, and $0.005 \leq c \leq 0.05$ all train well.

C.1 Limitations

Though our blob representation allows for powerful unsupervised, disentangled scene representations, our model still suffers from various shortcomings. For example, trained networks struggle to disentangle smaller objects (*e.g.* lamps on desks), perspective from object shape, and, occasionally, foreground appearance from background. Further, as shown in the main paper, blobs display a predilection toward certain canvas regions, though whether this is an artifact of dataset bias or model design remains unclear.

Appendix D Comparison to previous work

In Figures 20 and 21, we show random samples of untruncated images before and after style swapping. At a given level of photorealism as measured by FID,

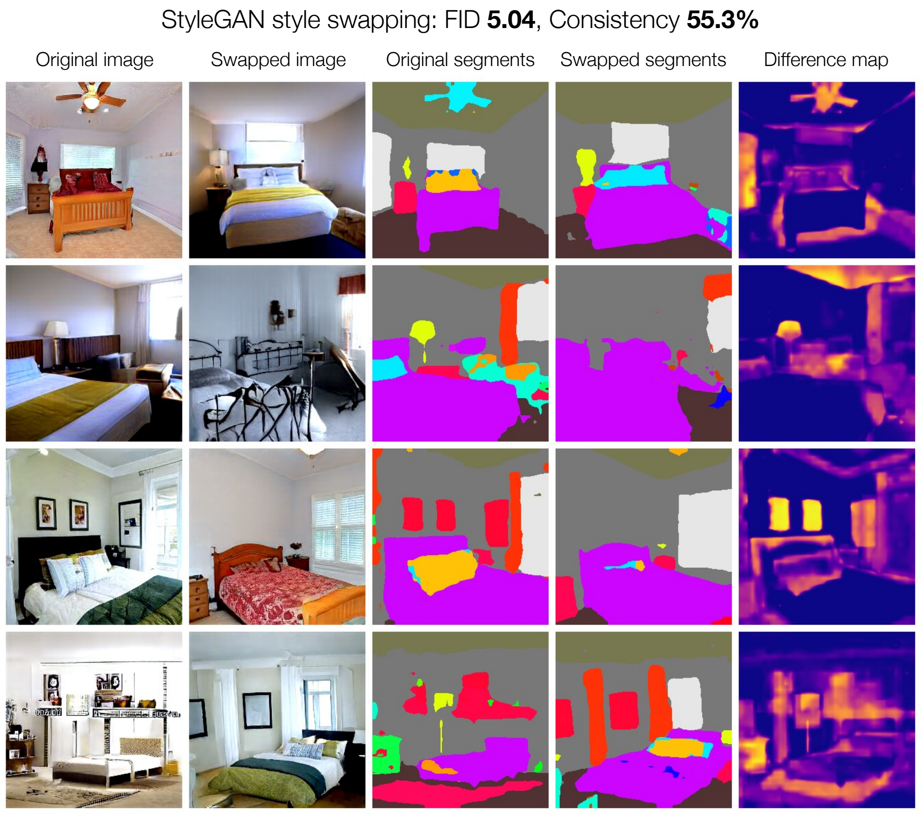


Fig. 20. Style swapping with StyleGAN: We show randomly sampled untruncated StyleGAN images before and after style swapping at layer 4, attaining an FID of 5.04 and layout consistency of 55.3%. The difference map shows the normalized KL divergence of the predicted per-pixel logits before and after swapping.

our model is able to produce layouts far more consistent with the original image thanks to its disentangled, compositional representation.

Lastly, we visualize the trade-off between the precision and recall metric [42] as we change the truncation value in Figure 24. Our model generates more perceptually realistic images than StyleGAN at all truncation values $0.0 \leq w \leq 1.0$, although the maximal recall at $w = 1.0$ is lower. In particular, our untruncated model performs better at both precision and recall than *all StyleGAN-generated images with* $w < 0.7$. These results provide evidence for the suggestion that our model's FID is higher because it cannot properly model outlier bedroom scenes using the blob representation.

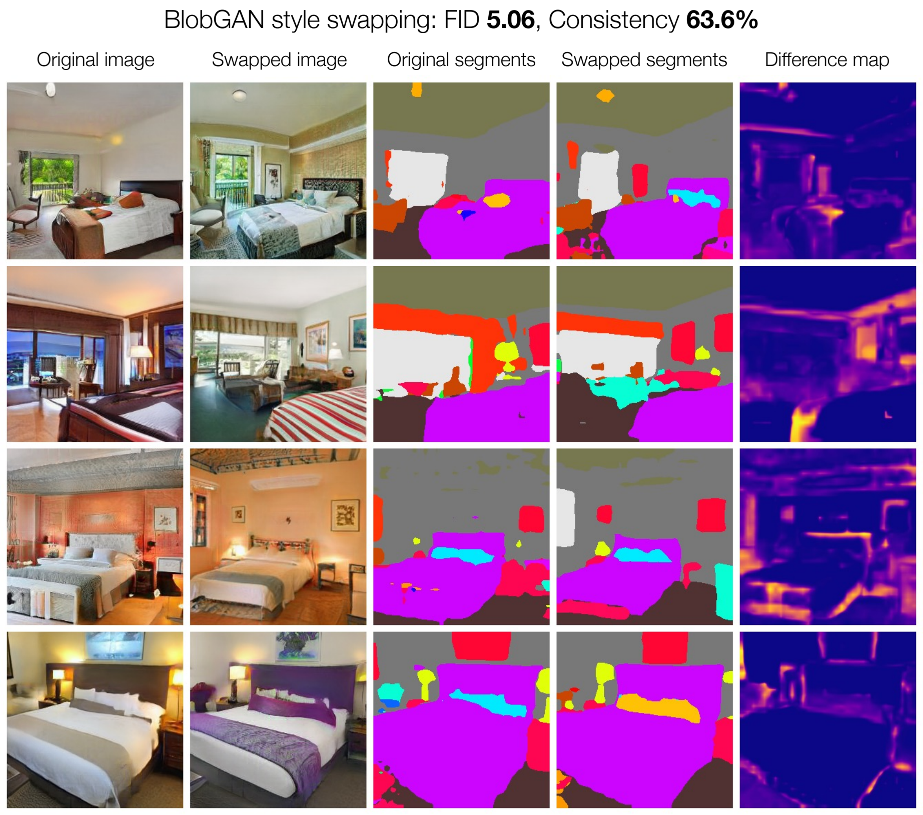


Fig. 21. Style swapping with BlobGAN: We show randomly sampled untruncated BlobGAN images before and after style swapping, attaining an FID of 5.06 and layout consistency of 63.6%. The difference map shows the normalized KL divergence of the predicted per-pixel logits before and after swapping.

Appendix E Model implementation

E.1 Hyperparameters and training

For the bedroom model trained in the paper, we use $d_{\text{in}} = 768$ and $d_{\text{style}} = 512$. Our generator with $k = 10$ blobs has 57.2 million parameters: 21.3 million in F and the remaining 35.9 million in G.

The model trained on LSUN conference rooms uses $k = 20$ and has 34.5M parameters in F; all other hyperparameters are as in the bedroom model.

The model trained on the union of LSUN kitchens, living rooms, and dining rooms uses $k = 45$ due to the increased complexity of the combined dataset, and thus reduces $d_{\text{in}} = 256$ and $d_{\text{style}} = 256$. This model has 61.3 million parameters in the generator: 31.3M in F and 30.0M in G.

We train all models for 1.5 million gradient steps with batch size 24 per-GPU across 8 NVIDIA A100 GPUs, except the bedrooms models (both BlobGAN and StyleGAN2), which are trained for 2.8 million steps. On the bedrooms model, we

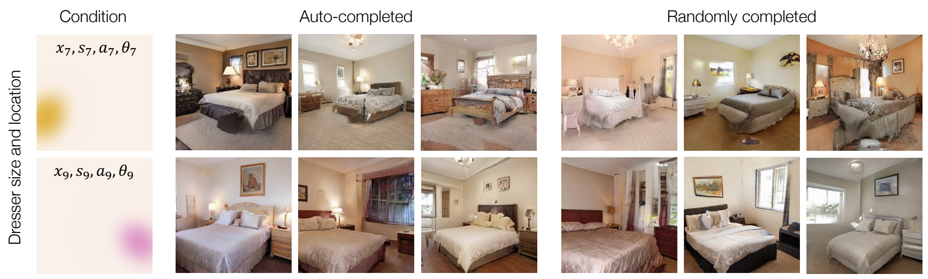

Condition
Auto-completed
Randomly completed
Dresser size and location
x_7, s_7, a_7, θ_7
x_9, s_9, a_9, θ_9

Fig. 22. Scene auto-complete: Various conditional generation problems fall under the umbrella of "scene auto-complete", *i.e.* optimizing random noise vectors to match a set of blob parameters when run through our layout network F. We show prediction of plausible scenes given the location and size (but not style) of dressers and nightstands. The model must not only predict the arrangement of the missing blobs, but also assign all blobs realistic appearance. When sampling target images randomly, objects are often randomly inserted, removed, reoriented, or otherwise disfigured due to incompatibility.

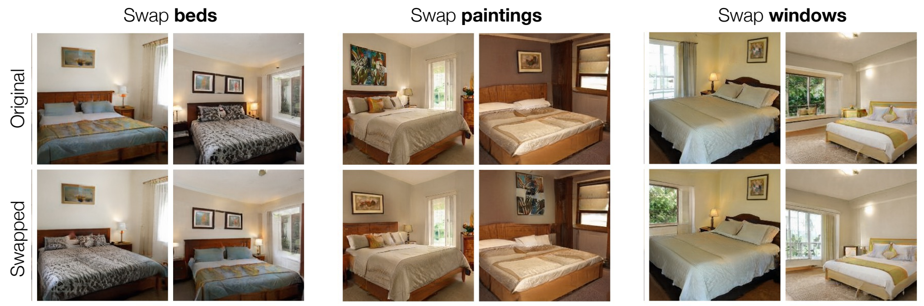


Fig. 23. Swapping blob styles: Interchanging ψ_i vectors without modifying layout leads to localized edits which change the appearance of individual objects in the scene.

experiment with $k = 20$ blobs as well as $k = 10$ blobs with no jitter. We find that results are less interpretable with 20 blobs and disentanglement is lower, perhaps since the model can "approximate" slightly higher-frequency data by using more blobs. This model also has a worse FID of 3.73. We also train a model with $k = 10$ blobs and no jitter, which attains comparable FID to the model with jitter, but with slightly reduced editing capabilities. Across all experiments, we find that changing k minorly impacts FID. Extra blobs mostly go unused, but too few blobs mean objects cannot be properly separated.

E.2 Image sampling

We sample all images shown in the paper and Supplementary Material with truncation. We truncate latents at the penultimate layer of F, since truncating in blob parameters space leads to undesirable behavior (*e.g.* biasing blob coordinates toward the center of the image). Then, truncation of random noise vector z's output of blob parameters $\boldsymbol{\beta}$ with a weight of w gives:

$$\boldsymbol{\beta}_{\text{trunc}} = F_L\left((1-w) \mathop{\mathbb{E}}_{z' \sim \mathcal{N}(0,I)} [F_{0:-1}(z')] + wF_{0:L-1}(z)\right) \tag{6}$$

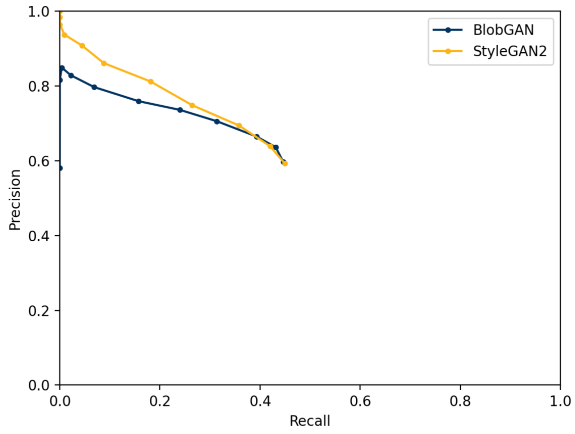


Fig. 24. We plot the precision-recall curve, by varying truncation values w, on LSUN bedrooms. Our untruncated model outperforms StyleGAN2 [39] with truncation values $w < 0.7$ in both precision and recall of generated images. While still outperforming StyleGAN2 on FID (Table 2), our model operates at a different point on this curve than StyleGAN2 – higher precision and lower recall – supporting the hypothesis that BlobGAN's FID suffers due to its inability to model long-tail, oddly-formed scenes.

Where $F_{l:m}$ represents layers l through m, inclusive, of the network which has L layers total. In practice, we approximate the expectation by sampling 100,000 random noise vectors. We use $w = 0.6$ or $w = 0.7$ for all bedroom images. $w = 0.5$ for images of conference rooms, and $w = 0.4$ for other indoor scenes, except when indicated otherwise ($w = 1$ means no truncation).

E.3 Object style swapping

When swapping styles between objects, rather than splatting the target (new) object's style $\psi_{i,\text{tgt}}$ directly onto the source (original) image's background style $\psi_{\text{bg, src}}$, we interpolate first between $\psi_{i,\text{tgt}}$ and then $\psi_{\text{bg,tgt}}$ (*i.e.*, the target image's background) at the border of the blob, and then splat this onto the background $\psi_{\text{bg, src}}$.

We find this necessary since the model learns to treat features on the border of a blob, which are typically a convex combination of the blob feature and the background feature, as belonging to the blob; when an unanticipated background feature becomes part of the feature along the border, the model is more prone to producing artifacts. This simple procedure mitigates this undesirable behavior and is trivially fully automated.

E.4 Spatial modulation

In StyleGAN2, convolution weights at layer l, $\theta_l \in \mathbb{R}^{d_l \times d_{l-1} \times k \times k}$, are multiplied by an affine-transformed style vector $w \in \mathbb{R}_l^d$ and then unit-normalized to perform modulation. Since our modulation varies spatially, we instead multiply input feature maps $x_{l-1} \in \mathbb{R}^{d_{l-1} \times h \times w}$ by a unit-normalized style grid $\Psi_l \in \mathbb{R}^{d_\text{style} \times h \times w}$ with a per-pixel affine transform, before convolving with unit-normalized weights θ_l to output new feature maps x_l. Affine transforms f map from d_style to d_l. More specifically, in StyleGAN2, modulated convolution is implemented as:

$$x_l = x_{l-1} * \frac{f(w_l) \odot \theta_l}{\|f(w_l) \odot \theta_l\|_2} \tag{7}$$

Since our styles are spatially varying, we cannot multiply convolution weights by the same broadcasted tensor throughout, and must modify our modulation:

$$x_l = \left(x_{l-1} \odot \frac{f(\Psi_l)}{\|f(\Psi_l)\|_2} \right) * \frac{\theta_l}{\|\theta_l\|_2} \tag{8}$$

We find this normalization scheme, also used in [59, 60], to work well in practice despite not having the same statistical guarantees as the original derivation.

E.5 Uncurated samples

In Figures 25 and 26, we show randomly sampled images from our model and StyleGAN2 trained on LSUN Bedrooms. We show the same on LSUN kitchens, living rooms, dining rooms, and conference rooms in Figures 27, 28, 29, and 30.

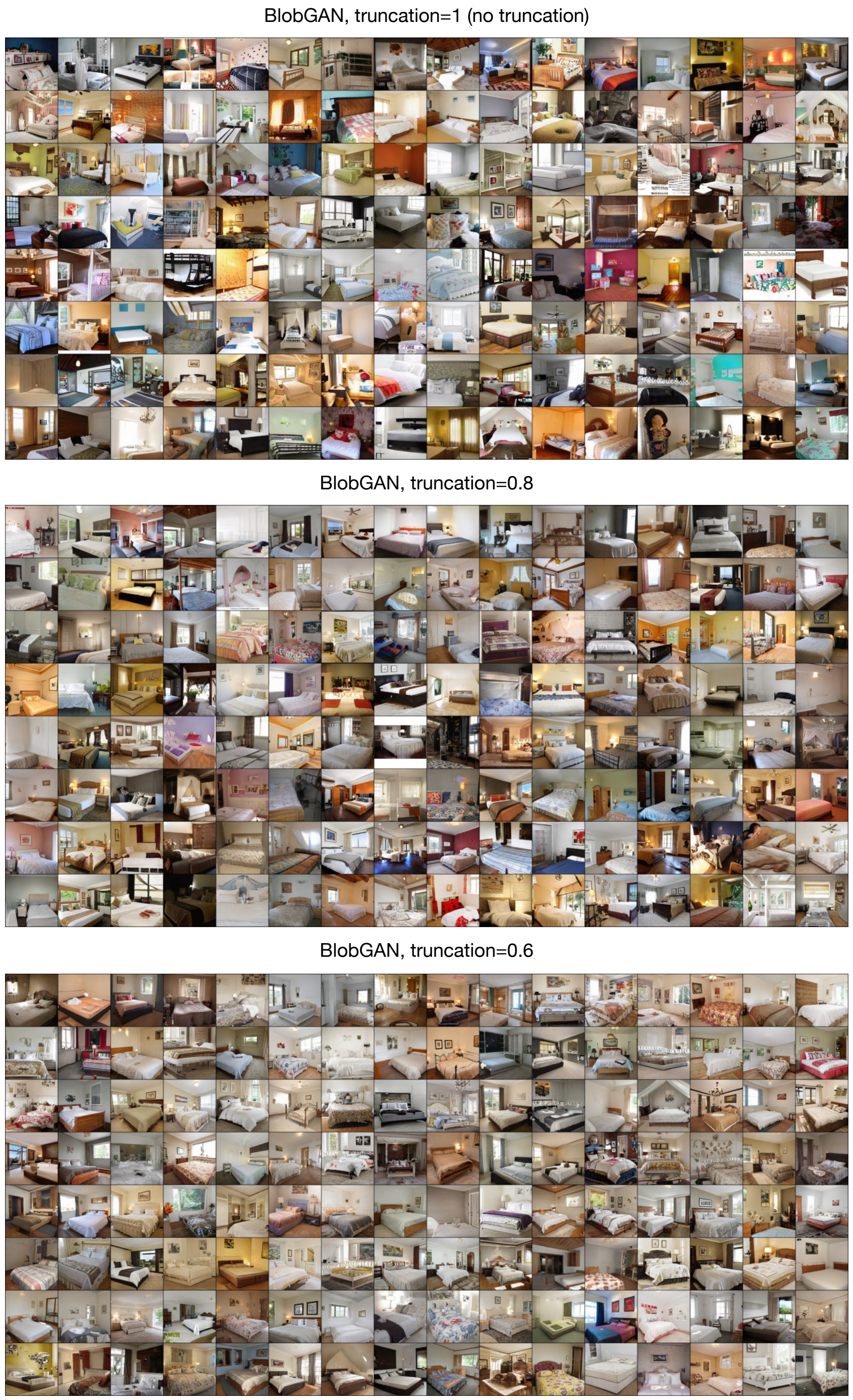


Fig. 25. We show uncurated random image samples from BlobGAN on LSUN bedrooms at various truncation levels. Please view zoomed in and in color for best results.

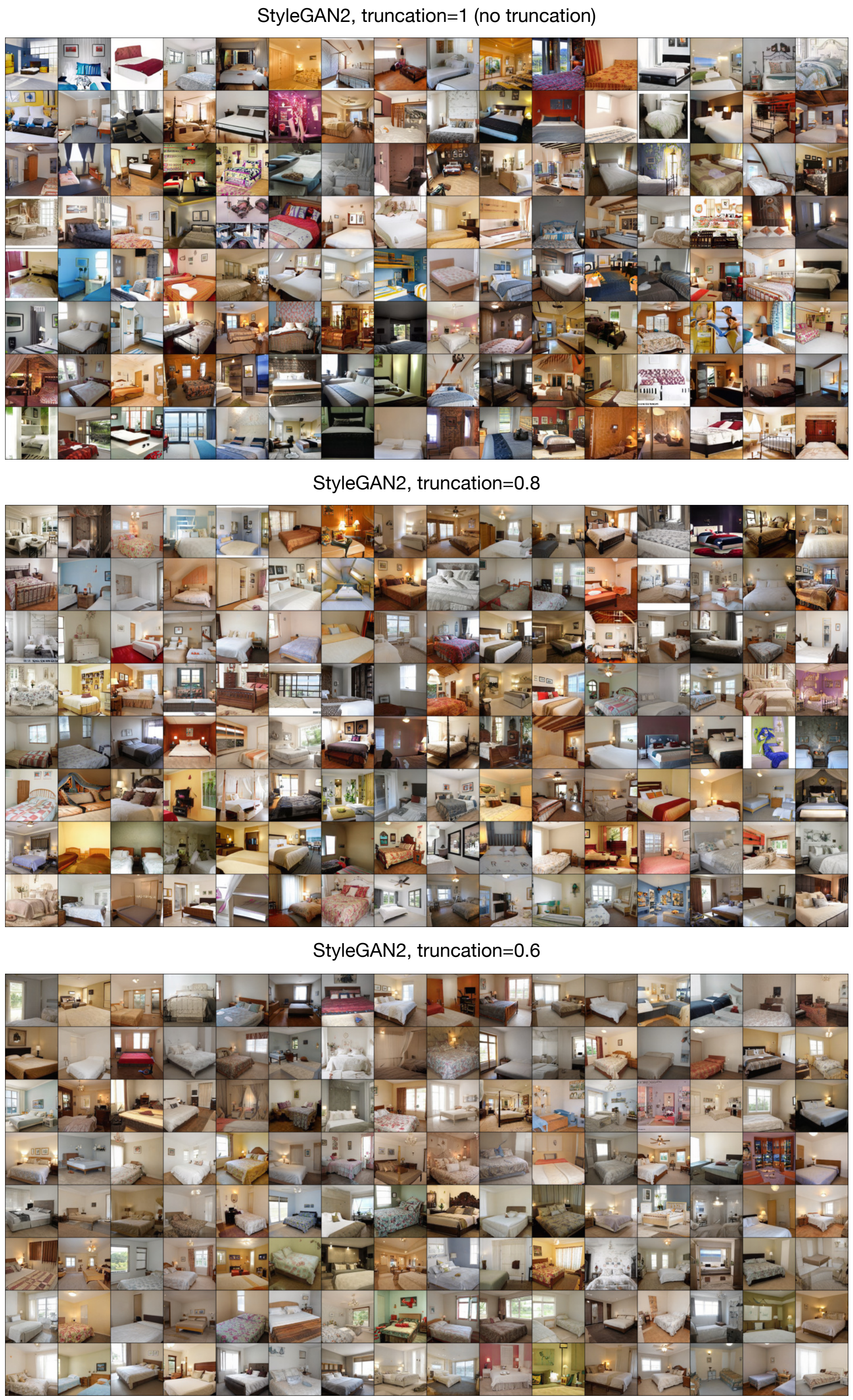
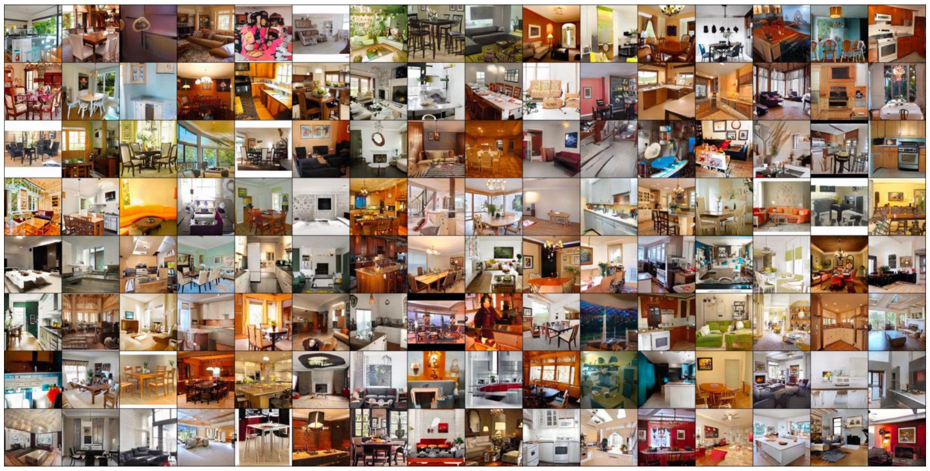
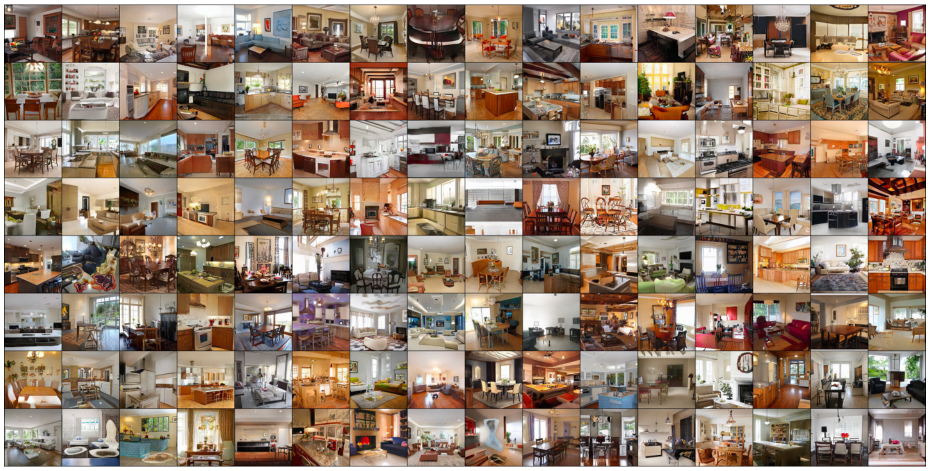

StyleGAN2, truncation=1 (no truncation)
StyleGAN2, truncation=0.8
StyleGAN2, truncation=0.6

Fig. 26. We show uncurated random image samples from StyleGAN2 on LSUN bedrooms at various truncation levels. Please view zoomed in and in color for best results.

BlobGAN, truncation=1 (no truncation)



BlobGAN, truncation=0.8



BlobGAN, truncation=0.6

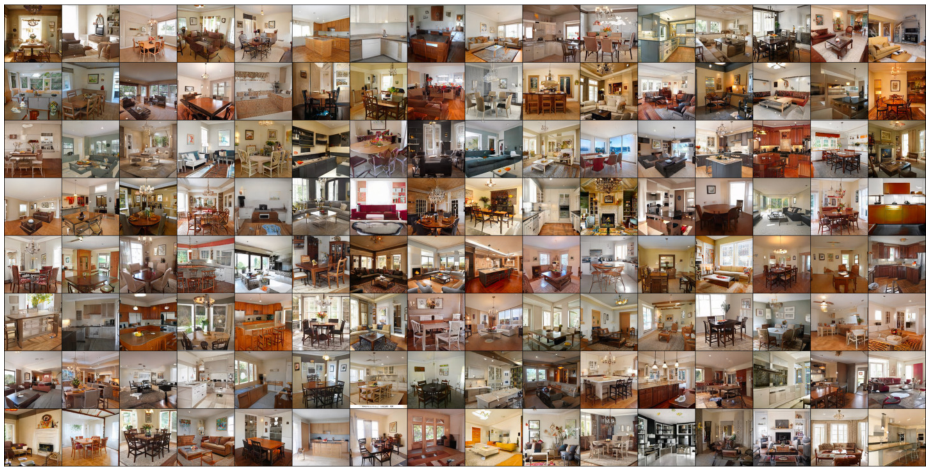


Fig. 27. We show uncurated random image samples from BlobGAN on LSUN kitchens, living rooms, and dining rooms at various truncation levels. Please view zoomed in and in color for best results.

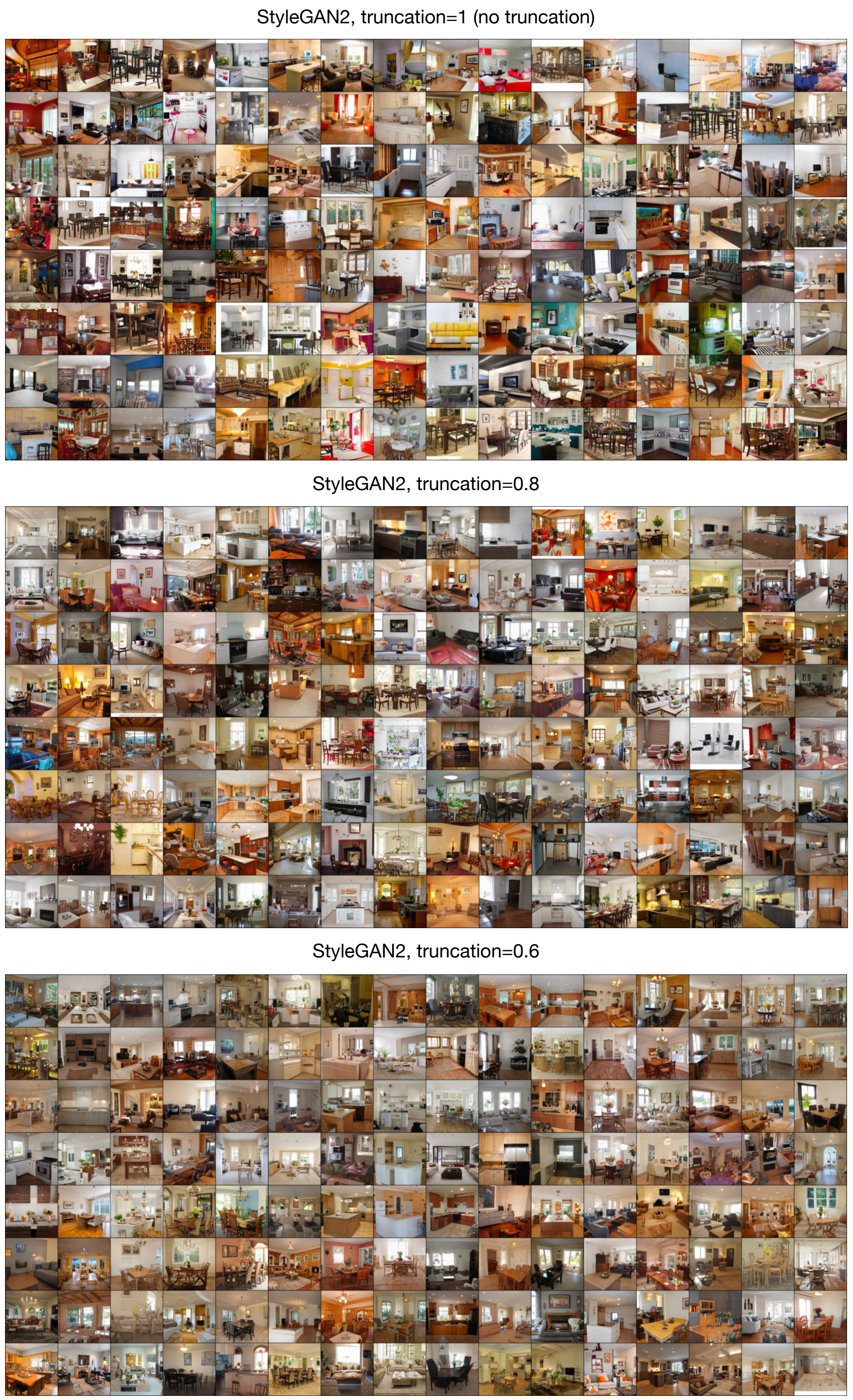

StyleGAN2, truncation=1 (no truncation)
StyleGAN2, truncation=0.8
StyleGAN2, truncation=0.6

Fig. 28. We show uncurated random image samples from StyleGAN2 on LSUN kitchens, living rooms, and dining rooms at various truncation levels. Please view zoomed in and in color for best results.

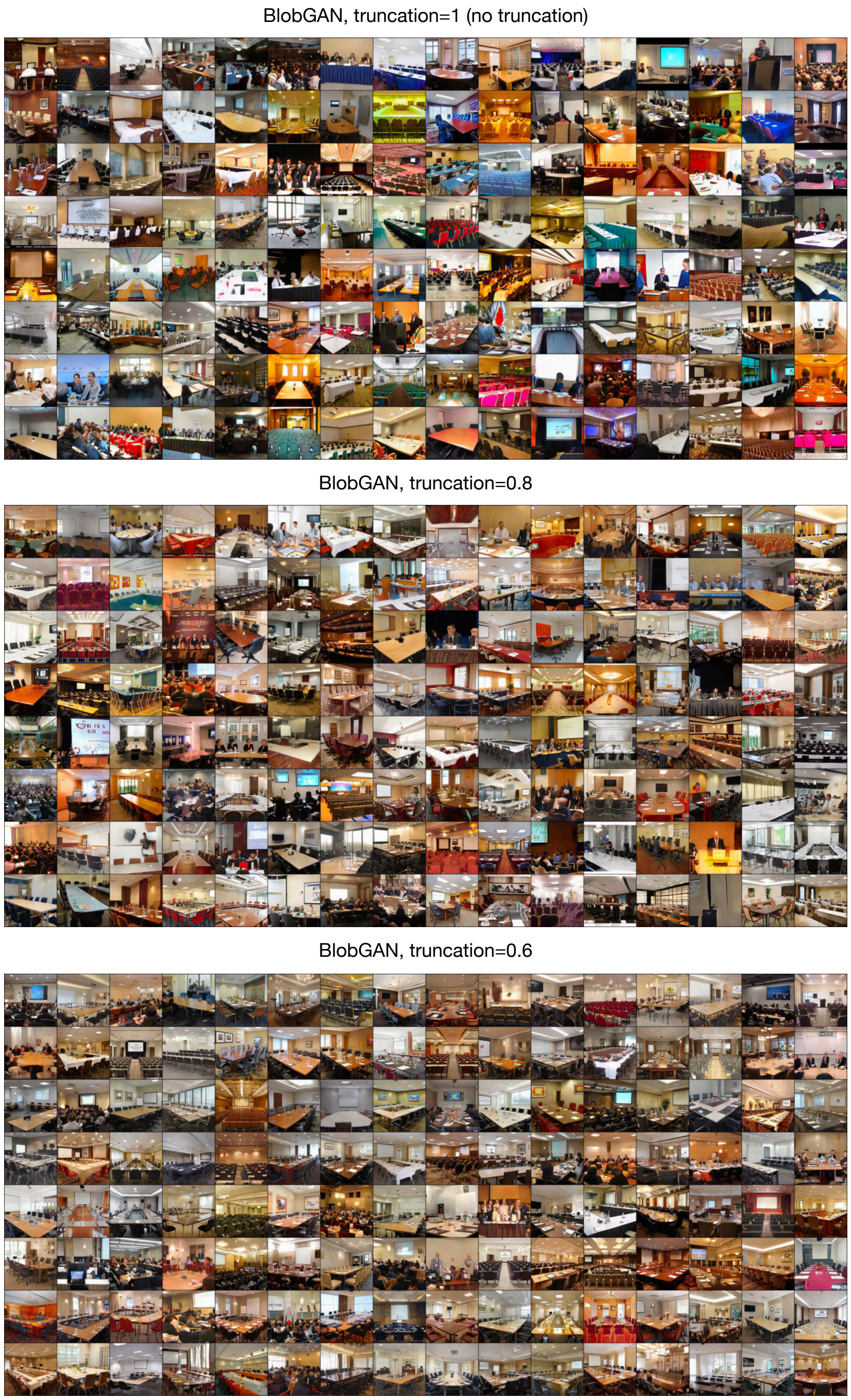


Fig. 29. We show uncurated random image samples from BlobGAN on LSUN conference rooms at various truncation levels. Please view zoomed in and in color for best results.

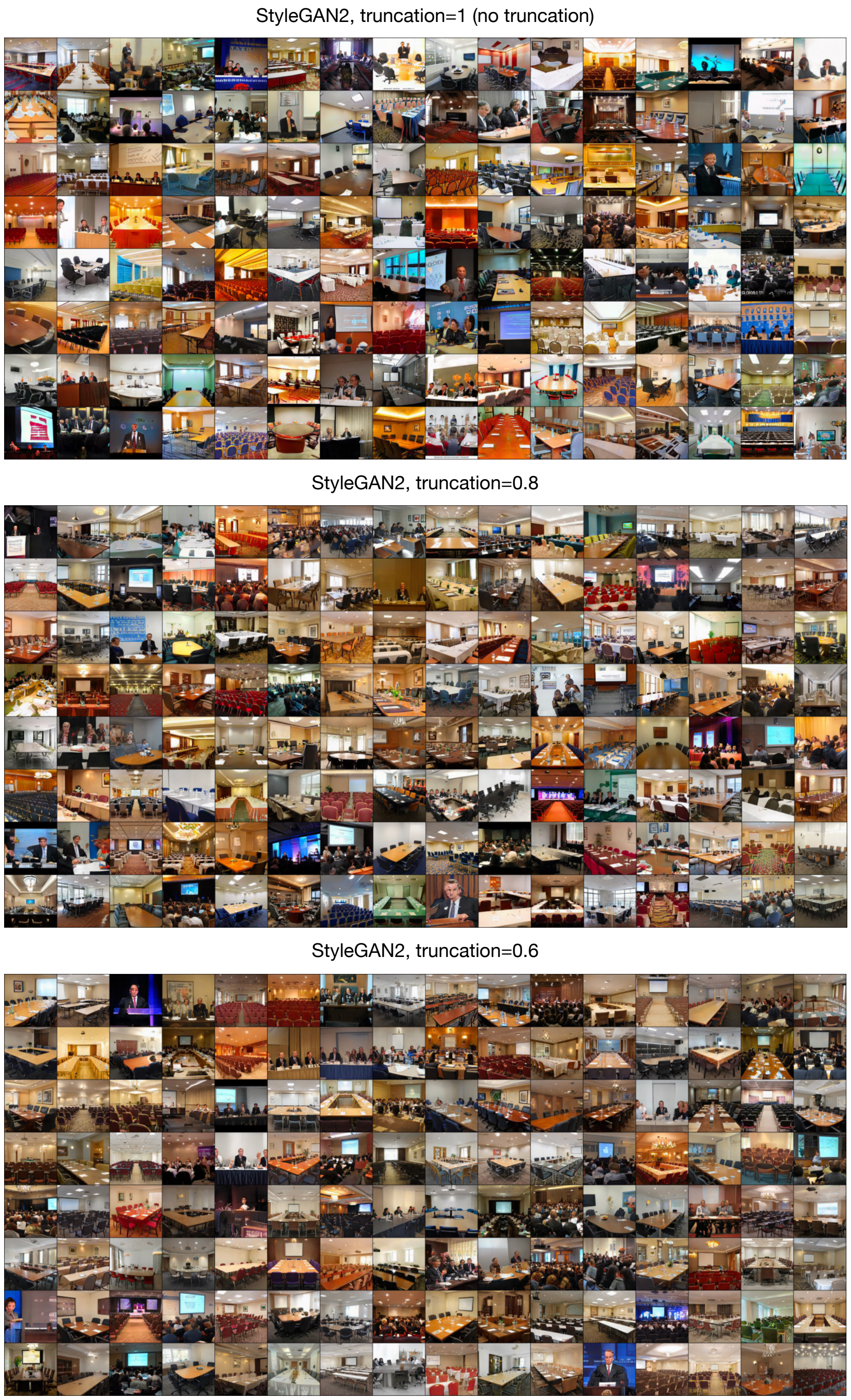

StyleGAN2, truncation=1 (no truncation)
StyleGAN2, truncation=0.8
StyleGAN2, truncation=0.6

Fig. 30. We show uncurated random image samples from StyleGAN2 on LSUN conference rooms at various truncation levels. Please view zoomed in and in color for best results.