Supplementary Material of

Multi-attribute Pizza Generator: Cross-domain Attribute Control with Conditional StyleGAN

Fangda Han[1]
fh199@cs.rutgers.edu
Guoyao Hao[1]
gh343@scarletmail.rutgers.edu
Ricardo Guerrero[2]
r.guerrero@samsung.com
Vladimir Pavlovic[1]
vladimir@cs.rutgers.edu

[1] Rutgers University
Piscataway, NJ, USA
[2] Samsung AI Center
Cambridge, UK

1 Visual Comparisons with Baselines

We provide additional samples from the proposed *MPG* and other state-of-the-art models to further prove the effectiveness of our model:

- Fig. 1(a): Example images generated using our *MPG*.
- Fig. 1(b): Example images generated using StackGAN2 [4].
- Fig. 1(c): Example images generated using CookGAN [1].
- Fig. 1(d): Example images generated using AttnGAN [3].

2 Traversing Attributes

As shown in Fig.1 of our main paper, the input of *MPG* is a triplet: food content (ingredients), geometric style (view attributes including view point, scale, horizontal and vertical shift), and visual style (diversity in fine-grained visual appearance of ingredients and the final dish). Here we provide examples of fixing one attribute group, and traversing through the other two groups. These examples provide additional qualitative evidence of the independence between the attribute groups and the smoothness of feature space learned by *MPG*.

- Fig. 2: Fix visual style and traverse ingredients and view attributes.
- Fig. 3: Fix view attributes and traverse ingredients and visual style.
- Fig. 4: Fix ingredients and traverse view attributes and visual style.

(a) *MPG*

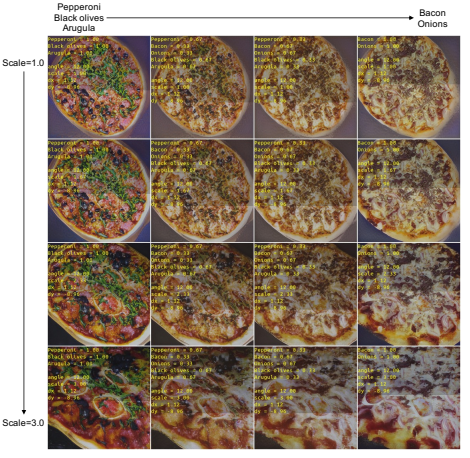
(b) StackGAN2 [4]

(c) CookGAN [1]

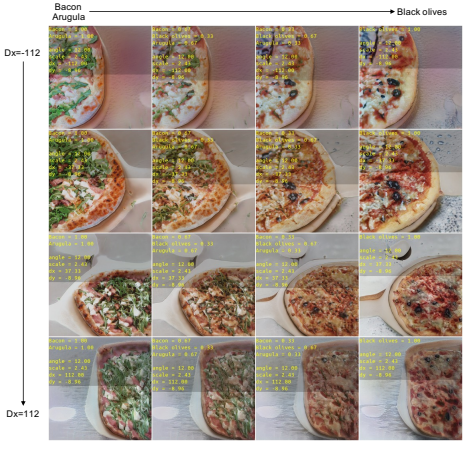
(d) AttnGAN [3]

Figure 1: Visual comparisons with baselines. View attributes and style noise are randomly sampled, with the ingredients and view attribute values shown in the top left corner of each image. Note that images generated by *MPG* have better quality than the baselines, with the desired ingredients more prominently displayed, without the visual distortions present in the competing approaches.

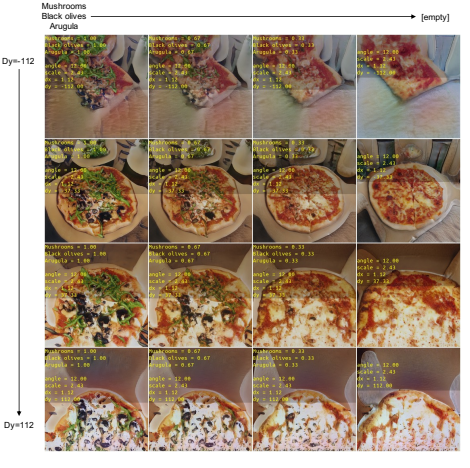
(a) ingredients (→) and angle (↓)



(b) ingredients (→) and scale (↓)



(c) ingredients (→) and dx (↓)



(d) ingredients (→) and dy (↓)

Figure 2: Images generated using *MPG* by fixing the visual style and traversing through **ingredients** (along horizontal axis) and a **single view attribute** (along vertical axis, with the other three view attributes fixed). Observe that in each sub-figure, pizza coloring, shapes, and image backgrounds are consistent in response to the desired, fixed visual style. The ingredients and the view attributes smoothly change as we gradually adjust the changing attributes from one end of the range to the other. For instance, in (a) the viewing angle of the pizza varies smoothly from top to bottom, but each column has the same toppings, starting from tomatoes and basil on the left to the pizza with no toppings on the right.

Figure 3: Images generated using our *MPG* by fixing the view attributes and traversing the **ingredients** (along horizontal axis) and the **visual style** (along vertical axis). Note that the visual style changing from z_1 to z_2 leads to a rotated, reddish pizza image turning elliptical for some z, while our four view attributes (view point, scale, shift) remain the same. The consistency is retained when we change the ingredients, across columns.

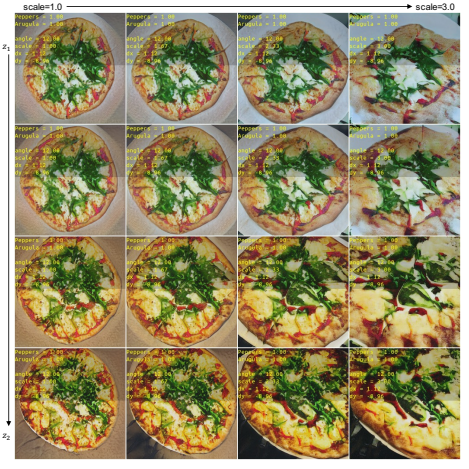
3 Dependency and Disentanglement of Attributes

We seek to quantitatively verify the ability of *MPG* to independently and effectively control the sets of attributes in synthesized images. To do so, we perform a correlation analysis among all pairs of attributes in images synthesized by *MPG*. Namely, we first select one attribute as the controlling (input) attribute and then examine (a) whether that attribute is correctly depicted in the image and (b) whether all other attributes are not affected by the controlling attribute. As the proxy for the assessed attribute, we use the output of the learned regressor / classifier for that attribute.

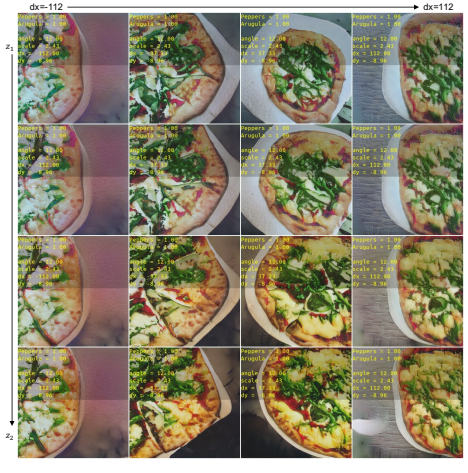
Fig. 5 depicts the results of this analysis. Each column corresponds to the traversal of one specific controlling (input) attribute; each row corresponds to the choice of a predicted attribute. For example, to plot the prediction of Bacon in synthetic images when controlling Pepperoni as the input attribute (row 2, column 1), we gradually increase Pepperoni input x_i from zero to one (ingredient labels are treated as continuous values as in the main paper). At each input value x_i, we generate 24 synthetic images and gather the corresponding Bacon ingredient classifier outputs $\{y_i^{(1)}, y_i^{(2)}, ..., y_i^{(24)}\}$. Blue points in each plot are the the scatter plot of x vs. y. After traversing through all x_is, we also compute the correlation coefficient between the controlling attribute 'Pepperoni' x and the predicted outputs 'Bacon' y and plot the trend line.

Focusing on column one (i.e., controlling Pepperoni), we can observe that as we increase the Pepperoni value in the *MPG* input (the "amount" of pepperoni), the prediction of Pepperoni from the synthetic image increases (row 1, column 1), indicating our ability to effectively control Pepperoni attribute. We also notice the prediction of Bacon (row 2, column 1) is almost a flat line (uniform scatter), which indicates that the existence of Bacon in *MPG* image is not influenced by the changing Pepperoni value in the input. Similar flat lines / uniform scatter also appear across other rows in column 1, suggesting that Pepperoni attribute is

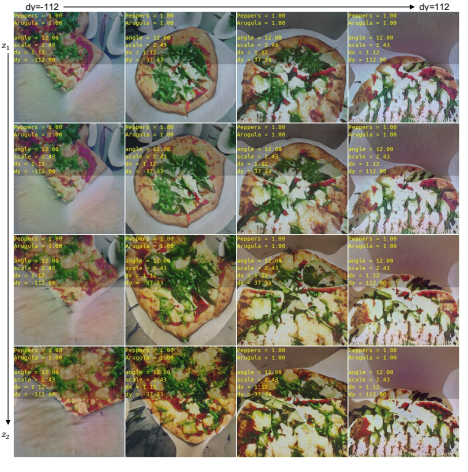
(a) angle ($\rightarrow$) and visual style ($\downarrow$)



(b) scale ($\rightarrow$) and visual style ($\downarrow$)



(c) dx ($\rightarrow$) and visual style ($\downarrow$)



(d) dy ($\rightarrow$) and visual style ($\downarrow$)

Figure 4: Images generated using our *MPG* by fixing the ingredients and traversing through **one view attribute** (along horizontal axis) and **visual styles** (along vertical axis). Notice that the ingredients are consistent and constant as we change the view attributes and the visual style. The visual style can be retained when changing the view attributes, *e.g.* the cutting marks from z_1 appear at the first row of each sub-figure and disappear as we move to z_2.

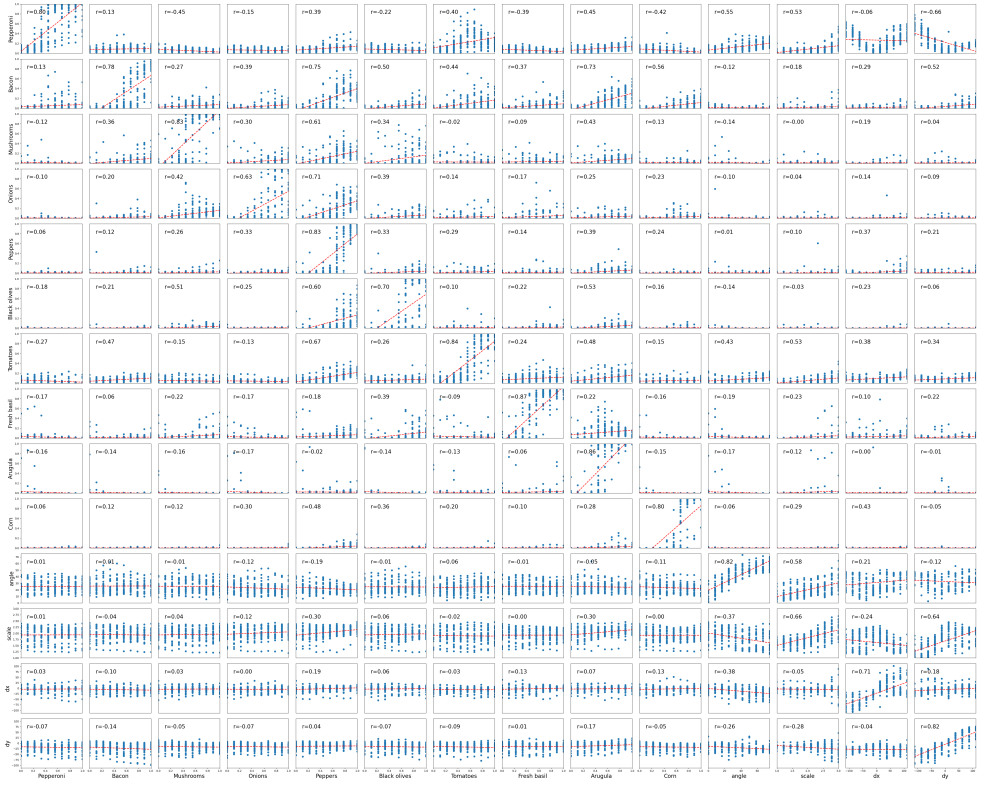


Figure 5: Dependency analysis of the predictions of an attribute, from synthesized images, (vertical axis in each plot) as a function of the controlling input attribute to *MPG* (horizontal axis of each plot). Each plot corresponds to a pair of (controlling input, predicted output) attributes. For instance, the top row of plots has the Pepperoni attribute as the predicted output, while the last column of plots has the vertical shift (dy) as the controlling input. Each plot also lists the correlation between the input and the output in the upper left corner. The red lines show the linear input-output trends (best fit). As expected, the diagonal plots, where the controlling input and the predicted output correspond to the same attribute, show strong correlation (diagonal trend), indicating the ability to control those attributes in the synthetic images. On the other hand, the off-diagonal plots show significantly lower input-output correlation (uniform scatter). This indicates that the predicted attribute is disentangled from (independent of) the controlling input. See Sec. 3 for more details.

independent of and disentangled from other ingredients as well as the view attributes.

The same observation holds for other (control,predict) pairs of ingredients and view attributes. The off-diagonal plots generally have smaller correlation coefficients compared with diagonal ones (see also the correlation heat map in Fig. 6), demonstrating the independence between attributes and the ability to control them effectively.

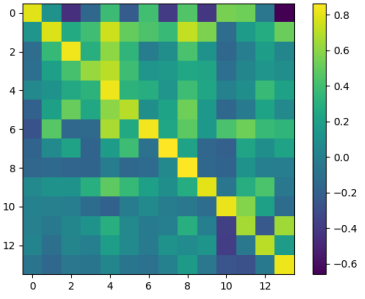


Figure 6: Heat map of correlation coefficients between each pair of (controlling input, predicted output) attributes, corresponding to numeric values of the correlation in the plots of Fig. 5. The diagonal displays stronger correlation, as desired, from that in the off-diagonal pairs

Figure 7: Two examples from SeFa [2]. For each example, three largest eigen values as well as images moving along the corresponding eigen vectors are displayed (from top to bottom)

4 Comparison with SeFa on Attribute Control

To verify *MPG*'s ability to control ingredients and view attributes, we implement SeFa [2] using pretrained *MPG* model in Fig. 7. SeFa is an eigen-decomposition-based method for finding the most significant attribute directions in unconditional GAN models. We clone their official code[1], load our pretrained *MPG* and visualize the result in Fig. 7. The source images (i.e., the first column of each example) are generated from random ingredients and commonly seen view attributes, followed by traversal along the eigen-directions corresponding to the three largest eigen values of 'all' layers (which performs better than those of the bottom, middle and top layers; refer to [2] for more details). The limitation of SeFa is threefold: (1) SeFa needs to manually correlate the desired attributes with the eigenvectors to infer the direction's meaning; (2) The largest eigenvectors do not always correspond to a single attribute, because of entanglement; (3) We are unable to find the eigenvectors that control the ingredient appearance.

[1] https://github.com/genforce/sefa

5 Assessment of Images with Uncommon View Attributes

We propose conditional FID and conditional mAP (i.e., c-FID and c-mAP) to verify the *MPG*'s performance at different view attribute values. To understand how c-FID and c-mAP are computed, we plot the predicted attribute value distributions along with the FIDs and mAPs of ingredients at different attribute values in Fig. 8. Taking the angle attribute as an example: the histogram is estimated using all samples in *Pizza10*; to compute FID and mAP at one specific angle value are computed, we fix the angle value and randomly sample other view attributes and ingredients to generate 5k fake images to estimate the fake image distribution and mAP, followed by the FID between the *fake distribution@5k* and *real distribution@Pizza10*. We observe that for each attribute, more frequent attribute values often result in improved FID and mAP, leading to better image quality and ingredients control.

c-FID (respectively c-mAP) of one attribute is computed by taking the average of the FIDs (respectively mAPs) at 10 evenly spaced values in the **targeting range** of the attribute. Since the estimated view attributes of real images in *Pizza10* are not uniformly distributed in the targeting range as shown in the frequency histograms in Fig. 8, we also compute c-FIDs and c-mAPs within three standard deviations around the mean value (i.e., **predicted range**) for each attribute (shown as *MPG* $_{3\sigma}$). The mean and standard deviation are estimated by fitting a Gaussian distribution from the frequency histogram.

We compare c-FID between *MPG* and its counterparts in Tab. 1. *MPG* exhibits better c-FID compared with those without *MSMAE*, which again affirms the effectiveness of MSMAE. Note that *MPG-AR* has the best c-FID; this is because without attribute regularizers, the model lacks sensitivity to view labels, able to generate reasonable images even at extreme view attribute values. Comparing c-mAP in Tab. 2, we observe that removing *MSMAE* again decrease c-mAPs. Both c-FID and c-mAP are further improved by setting the range to be the **predicted range**.

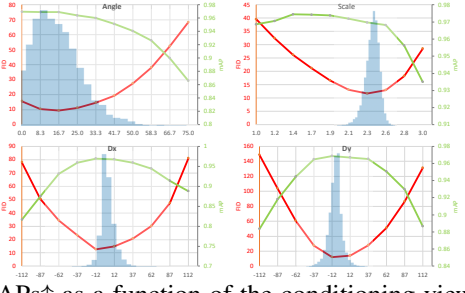


Figure 8: FIDs↓ and mAPs↑ as a function of the conditioning view attribute (angle, scale, dx, and dy). FIDs is estimated from 5k fake samples and all real images in *Pizza10*; mAP is estimated using 5k fake samples. The histograms depict the distribution of the predicted attribute values on *Pizza10* dataset

References

[1] Fangda Han, Ricardo Guerrero, and Vladimir Pavlovic. CookGAN: Meal image synthesis from ingredients. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1450–1458, 2020.

Models	c-FID$_{angle}$	c-FID$_{scale}$	c-FID$_{dx}$	c-FID$_{dy}$	avg c-FID
MPG-MSMAE	28.18	22.74	39.04	67.20	39.29
*MPG-MSMAE **	27.95	22.89	40.11	68.63	39.90
MPG-AR	**16.19**	**12.25**	**12.96**	**13.12**	**13.63**
MPG	26.75	22.07	39.26	66.34	38.61
MPG $_{3\sigma}$	20.21	15.98	14.56	15.39	16.54

Table 1: c-FID comparison between *MPG* and its counterparts with missing components, best values are depicted in **bold**

Models	c-mAP$_{angle}$	c-mAP$_{scale}$	c-mAP$_{dx}$	c-mAP$_{dy}$	avg c-mAP
MPG-MSMAE	0.9378	0.9611	0.9232	0.9204	0.9356
*MPG-MSMAE **	0.9496	0.9441	0.9133	0.9280	0.9338
MPG-AR	0.2043	0.1907	0.2161	0.2166	0.2069
MPG	0.9416	**0.9663**	0.9221	0.9377	0.9419
MPG $_{3\sigma}$	**0.9654**	0.9651	**0.9659**	**0.9664**	**0.9657**

Table 2: c-mAP comparison between *MPG* and its counterparts with missing components, best values are depicted in **bold**

[2] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *arXiv preprint arXiv:2007.06600*, 2020.

[3] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018.

[4] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. StackGAN++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1947–1962, 2018.