Jan. 13th, 2021

Class 5: Project

Exploring AI and Neural Nets in Design

Gia Jung
Research Associate, Lab for Design Technologies, Harvard University
Irving Innovation Fellow 2020 - 2021

Claire Djang
Lab for Design Technologies, Harvard University
Currently at Certain Measures

Housekeeping

1. Sign up for Friday final presentation time slot on this sign-up sheet !

2. Sign up for individual meeting/desk crit with Gia & Claire here !

5.1

Latent Space

A closer look at latent space and latent space vectors:

Grids

Iterations

Vector Arithmetic

5.2

Visualization Workshop

Grids

Interpolation with Grids

Integer Iterations

Vector Arithmetic

5.3

Individual Meetings

Progress meetings beginning at

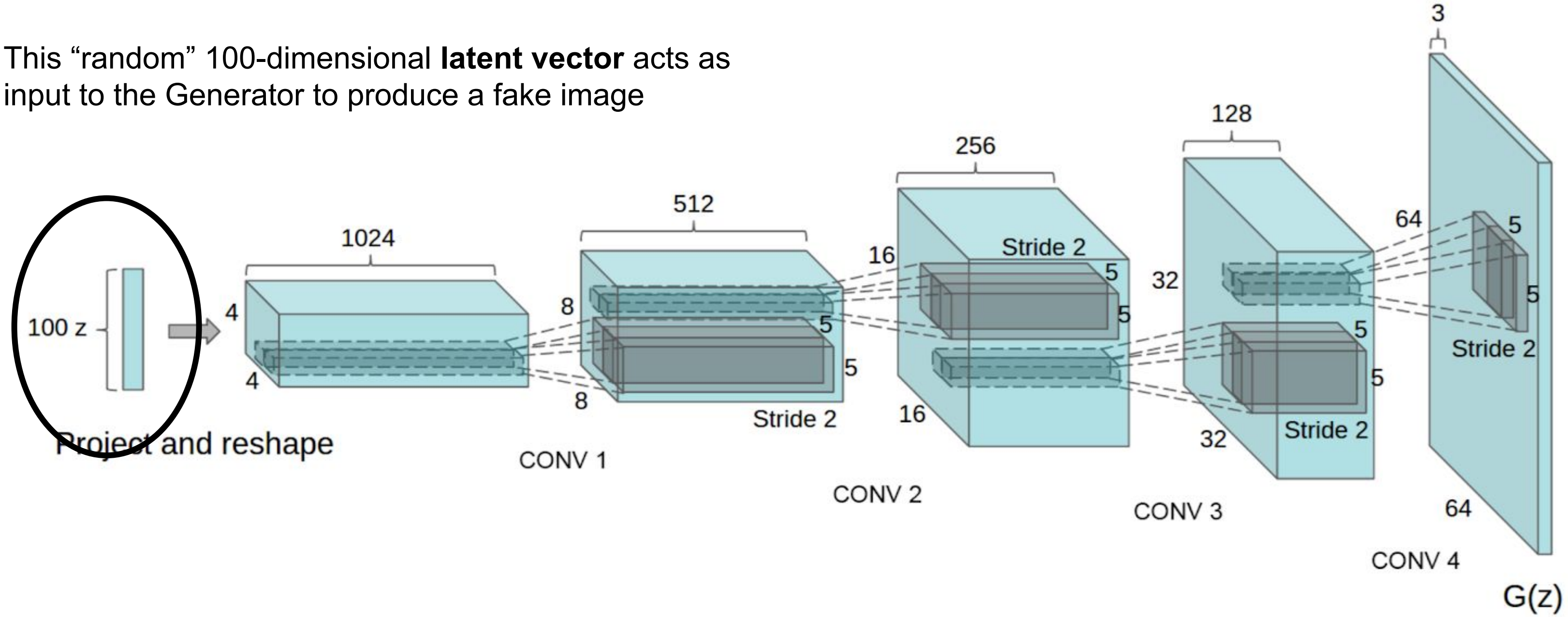
2:00pm EST

Data	**Model**	**Project**
1. Collection	1. Choosing a Model	Latent Space Exploration #1 **Interpolation Animation**
2. Curation	2. Training a Model	Latent Space Exploration #2 **Interpolated Grid**
3. Processing		Latent Space Exploration #3 **Vector Arithmetic**

What is Latent Space?

The Generator produces a fake image from the “noise vector”

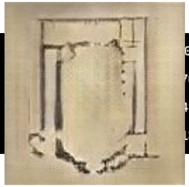
This “random” 100-dimensional **latent vector** acts as input to the Generator to produce a fake image



DCGAN Architecture (Generator)

From Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

Every unique fake image is mapped from a unique latent vector. We can regenerate the same image using the same vector.

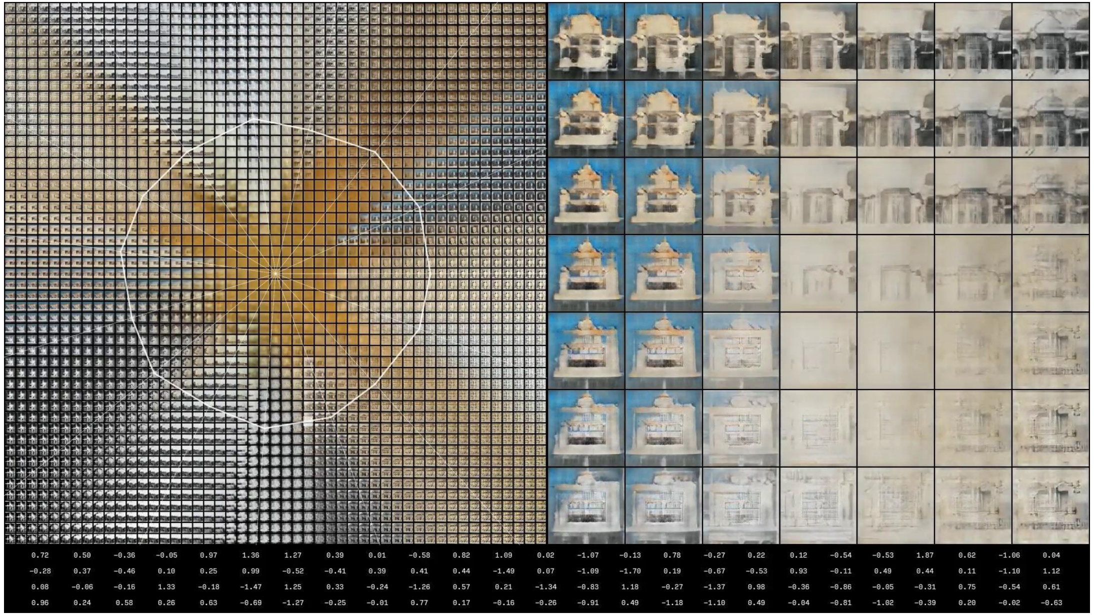


```
-0.86   0.39   1.67   1.02  -2.22   1.09  -0.99   0.21   0.10  -0.34  -0.28              0   1.24   0.26  -1.17  -0.71  -1.54  -2.73   0.87   0.79   1.15   1.02  -0.19
 1.03   0.57   1.02  -0.42  -0.03   0.79   1.51   0.83   0.05   0.58  -0.72             76  -0.09  -2.07   1.13   1.57   0.83   0.06  -1.41   1.45   0.71   0.52  -0.34
-0.41   0.91   0.80  -0.44   0.09  -0.37  -0.99   0.36  -0.66  -0.38  -0.56             41  -0.28   0.38  -0.70  -0.88   0.07   0.21  -1.20   0.59  -0.45  -2.68   1.33
 0.59  -0.93   0.86   0.79   0.76   0.65   0.93   0.01   1.68   0.28  -2.60             13   1.46  -1.49  -0.43   0.19  -0.37   0.52  -0.63   0.72  -0.38   1.06   0.55
```

```
 0.78   0.09  -0.06   0.35   0.26  -0.29   1.02   1.51  -0.56  -1.16   0.36              3   0.01   0.84   0.08   0.64   0.60   1.71   1.47  -0.34   0.85  -1.59   0.06
-0.68   0.58   0.72   1.09   0.47  -0.92   0.89   0.15   0.29   0.04  -0.61             79  -0.31   0.52  -0.23  -1.12   0.30  -0.91  -0.07  -0.46  -0.18  -0.14   0.64
-1.68   0.72   0.52   0.92  -0.25   0.88   0.84  -0.38  -1.77  -0.67   0.68             05  -0.19  -0.48  -0.36   0.18   0.33  -0.16   2.52  -0.43  -0.76  -0.09  -0.91
 0.87   0.26  -0.37   0.26  -0.76   1.39  -1.65   0.54  -0.25  -0.09  -1.20              8  -0.61   0.21  -1.15   0.45   0.91  -0.05  -0.15   0.07  -0.05   1.44  -0.10
```

```
 1.01   0.01  -0.30  -0.90  -0.58  -0.96   0.41   1.28   0.94   0.37   0.77              6  -0.03  -0.26  -1.87   0.90  -0.56   0.33   0.67   0.04  -0.44  -0.03   0.48
 0.70  -0.07  -0.19   0.39   0.19  -0.74   0.07  -1.01   1.20   0.77   0.00              8   0.10   0.44   0.32  -0.68  -0.91  -1.09  -0.09   0.35   0.12  -0.92  -0.16
 0.90  -2.21  -1.03  -1.08   0.68   1.28  -0.11  -0.27  -0.67  -0.24   0.66              1   1.15   0.52   0.25   0.81   0.05   0.27   0.30   0.70  -0.46   0.66  -1.08
 0.15   0.17   0.16  -0.61  -0.01  -0.40  -0.47   0.63   0.81   0.04  -0.47             04   0.91  -1.64   0.98  -0.23   0.45   0.43  -0.51   0.37  -1.07  -0.45  -0.43
```

Every unique fake image is mapped from a unique latent vector. We can regenerate the same image using the same vector.

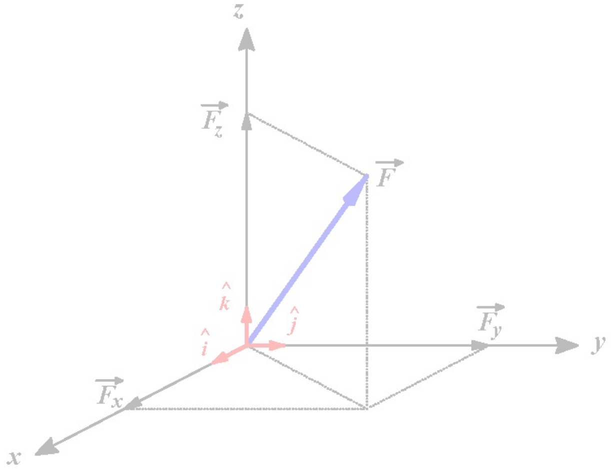


```
0.72   0.50  -0.36  -0.05   0.97   1.36   1.27   0.39   0.01  -0.58   0.82   1.09   0.02  -1.07  -0.13   0.78  -0.27   0.22   0.12  -0.54  -0.53   1.87   0.62  -1.06   0.04
-0.28  0.37  -0.46   0.10   0.25   0.99  -0.52  -0.41   0.39   0.41   0.44  -1.49   0.07  -1.09  -1.70   0.19  -0.67  -0.53   0.93  -0.11   0.49   0.44   0.11  -1.10   1.12
0.08  -0.06  -0.16   1.33  -0.18  -1.47   1.25   0.33  -0.24  -1.26   0.57   0.21  -1.34  -0.83   1.18  -0.27  -1.37   0.98  -0.36  -0.86  -0.05  -0.31   0.75  -0.54   0.61
0.96   0.24   0.58   0.26   0.63  -0.69  -1.27  -0.25  -0.01   0.77   0.17  -0.16  -0.26  -0.91   0.49  -1.18  -1.10   0.49  -0.04  -0.81  -1.02  -0.39   0.20  -0.02  -0.63
```

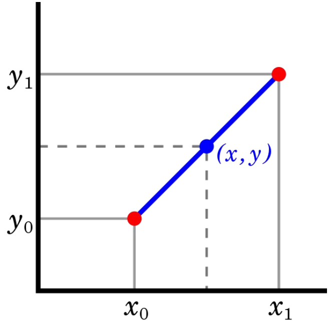
“Beaux-art Latent Walk” (2020)
Lab for Design Technologies
Prof. Andrew Witt, Gia Jung, Claire Djang

Think of Latent Space as a 100-dimensional Vector Space where Latent Vectors live

$$\text{vector} = (v_1, v_2, v_3, \ldots, v_{99}, v_{100})$$

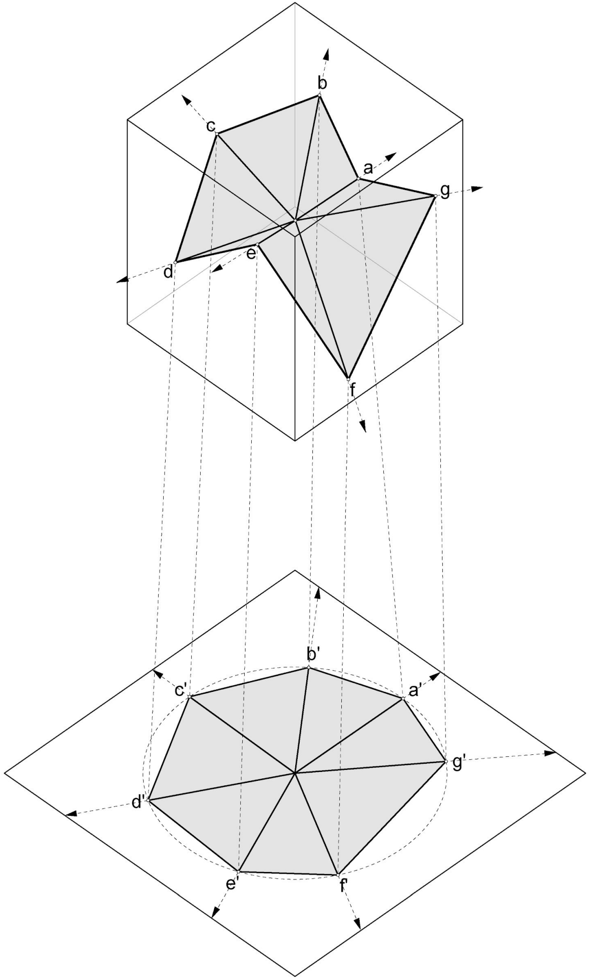


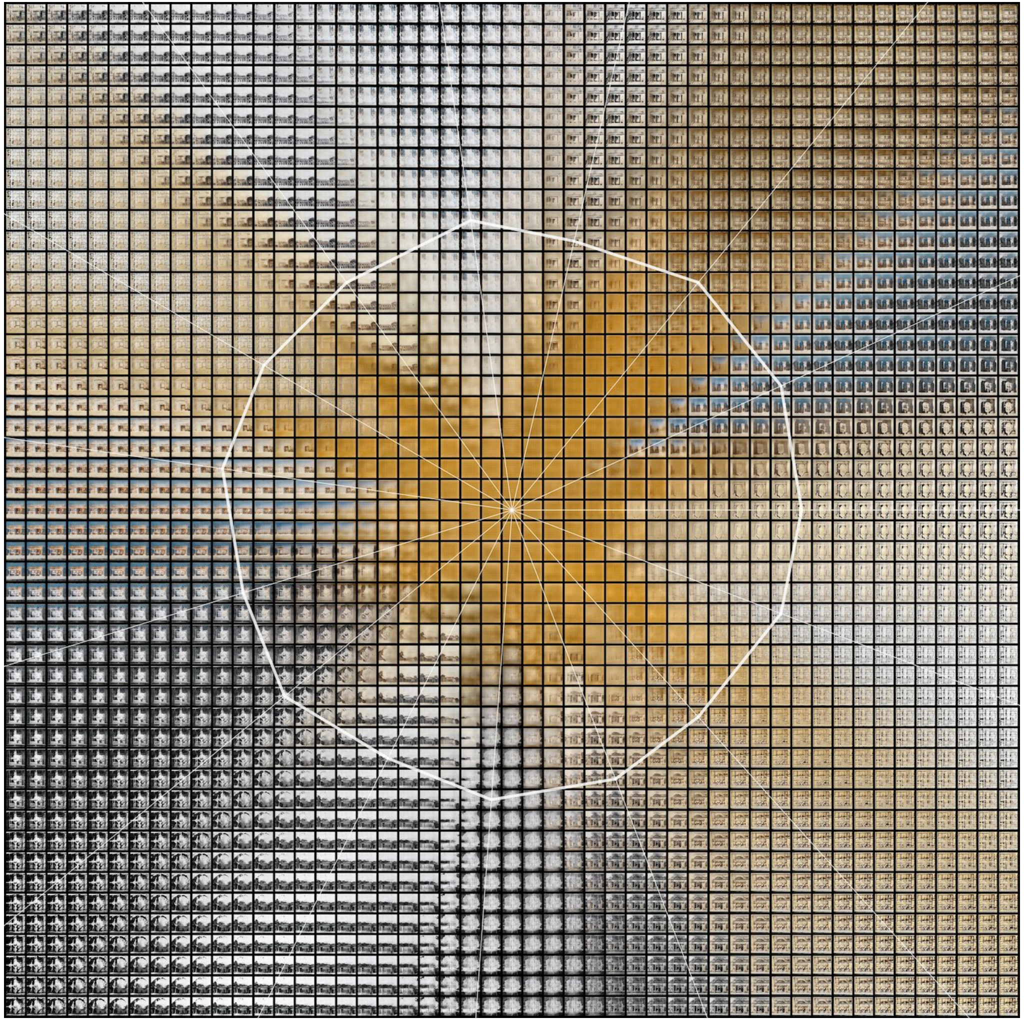
Linear Interpolation



Grids

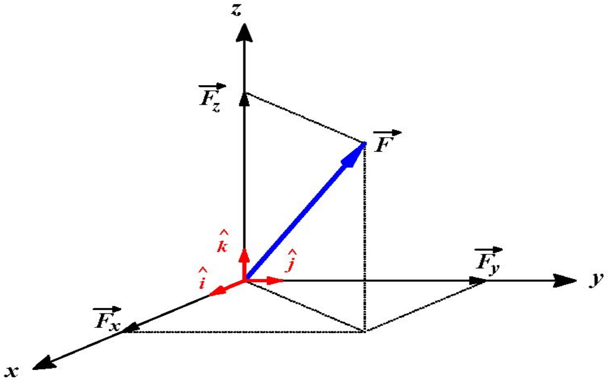
How can we explore the latent space nearby a chosen vector, as a way of understanding its local environment?

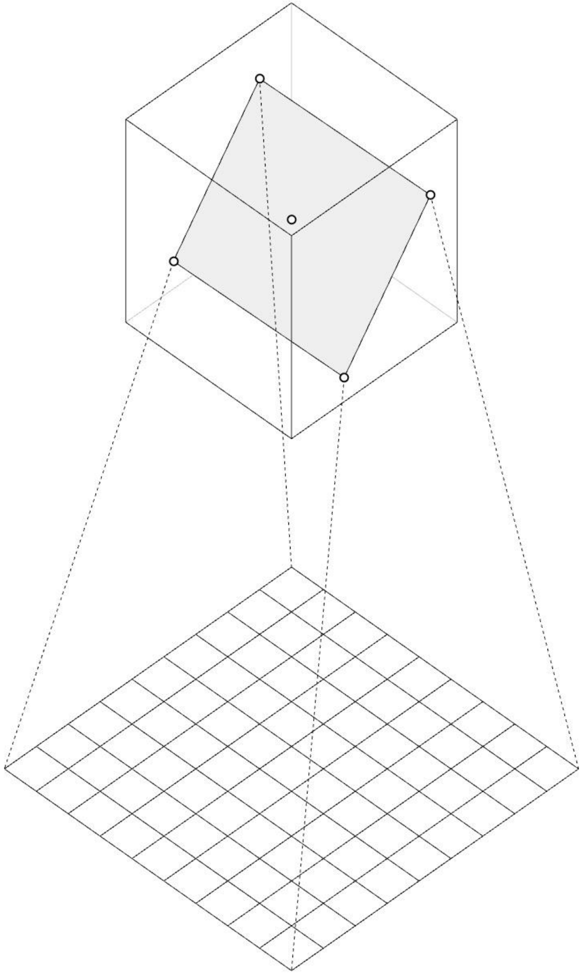

b
c
a
g
d
e
f
b'
c'
a'
g'
d'
e'
f'



3D analogy:
think of planar surfaces that cut through 3D space

$$\vec{F} = \vec{F_x} + \vec{F_y} + \vec{F_z}$$



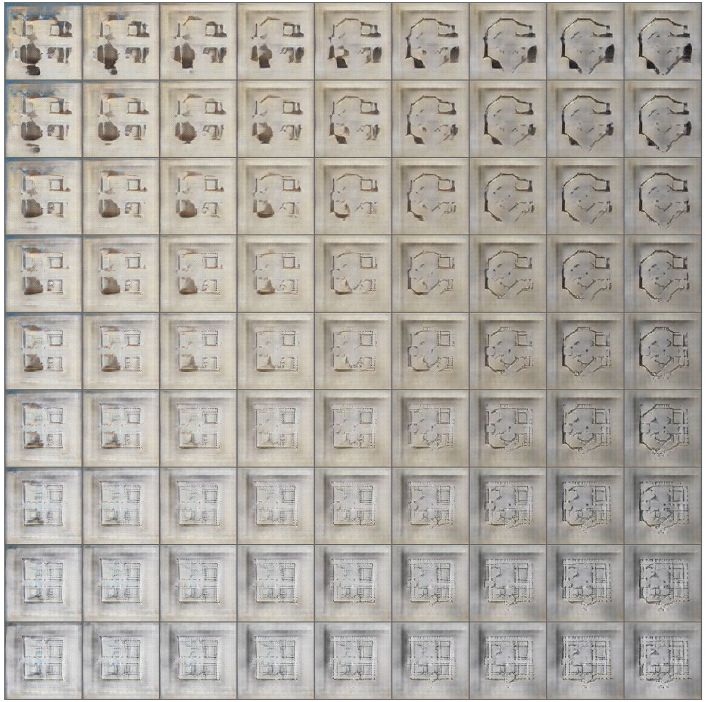


Smooth Grid

The central square is generated from a given 100-dimensional latent vector. It is surrounded by other generated images whose vectors have been displaced a fixed amount from the original latent vector.

The displacement for each dimension is between -1 and 1 with .25 increments.

The displacement can be represented by a 100-dimensional vector where each value is between -1 and 1 at .25 increments.

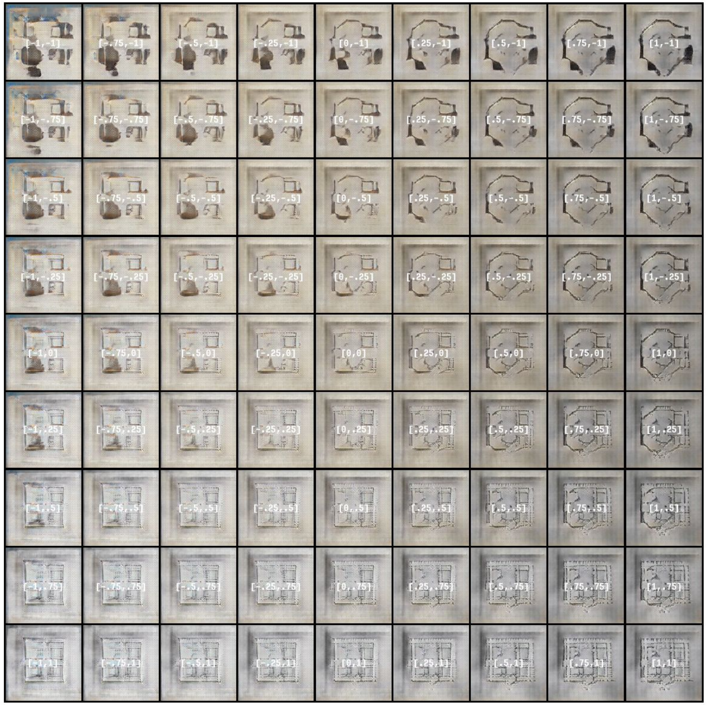


Smooth Grid

Reducing the latent
displacement vector into a
2-dimensional representation.
i.e. the vector below:

[0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0
1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1]

is represented as [0,1]
in the diagram.

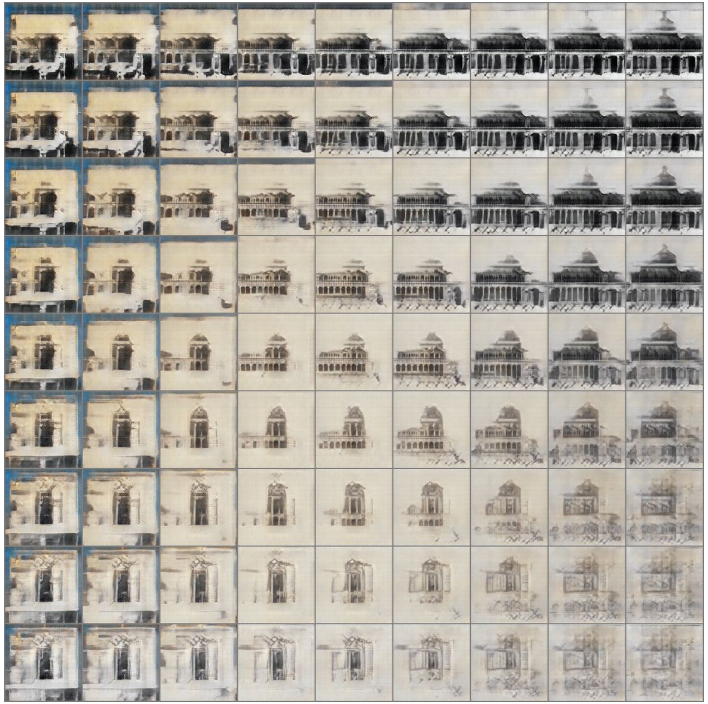


Smooth Grid

Reducing the latent displacement vector into a 2-dimensional representation. i.e. the vector below:
[0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0
1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1]
is represented as [0,1] in the diagram.

[-1,-1]	[-.75,-1]	[-.5,-1]	[-.25,-1]	[0,-1]	[.25,-1]	[.5,-1]	[.75,-1]	[1,-1]
[-1,-.75]	[-.75,-.75]	[-.5,-.75]	[-.25,-.75]	[0,-.75]	[.25,-.75]	[.5,-.75]	[.75,-.75]	[1,-.75]
[-1,-.5]	[-.75,-.5]	[-.5,-.5]	[-.25,-.5]	[0,-.5]	[.25,-.5]	[.5,-.5]	[.75,-.5]	[1,-.5]
[-1,-.25]	[-.75,-.25]	[-.5,-.25]	[-.25,-.25]	[0,-.25]	[.25,-.25]	[.5,-.25]	[.75,-.25]	[1,-.25]
[-1,0]	[-.75,0]	[-.5,0]	[-.25,0]	[0,0]	[.25,0]	[.5,0]	[.75,0]	[1,0]
[-1,.25]	[-.75,.25]	[-.5,.25]	[-.25,.25]	[0,.25]	[.25,.25]	[.5,.25]	[.75,.25]	[1,.25]
[-1,.5]	[-.75,.5]	[-.5,.5]	[-.25,.5]	[0,.5]	[.25,.5]	[.5,.5]	[.75,.5]	[1,.5]
[-1,.75]	[-.75,.75]	[-.5,.75]	[-.25,.75]	[0,.75]	[.25,.75]	[.5,.75]	[.75,.75]	[1,.75]
[-1,1]	[-.75,1]	[-.5,1]	[-.25,1]	[0,1]	[.25,1]	[.5,1]	[.75,1]	[1,1]

Smooth Grid

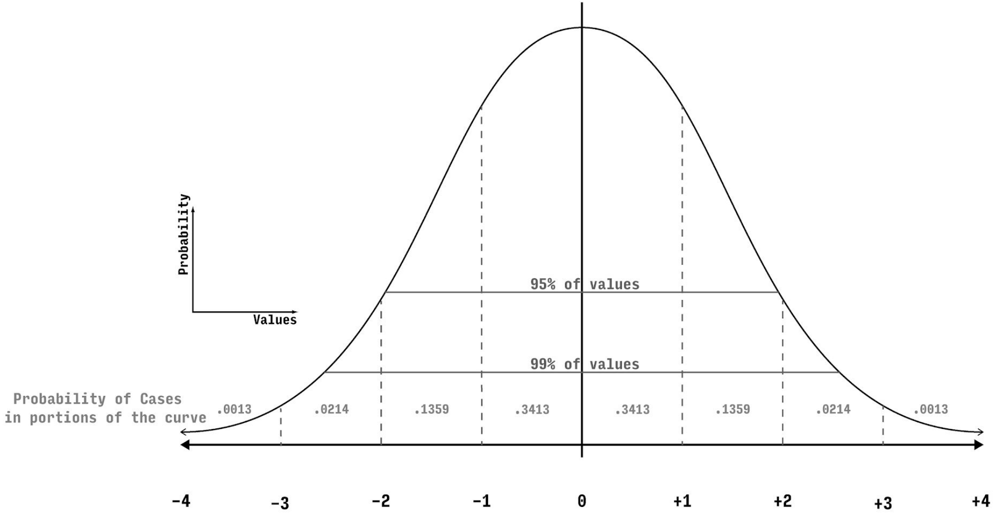


Smooth Grid

[-2,-2]	[-1.5,-2]	[-1,-2]	[-.5,-2]	[0,-2]	[.5,-2]	[1,-2]	[1.5,-2]	[2,-2]
[-2,-1.5]	[-1.5,-1.5]	[-1,-1.5]	[-.5,-1.5]	[0,-1.5]	[.5,-1.5]	[1,-1.5]	[1.5,-1.5]	[2,-1.5]
[-2,-1]	[-1.5,-1]	[-1,-1]	[-.5,-1]	[0,-1]	[.5,-1]	[1,-1]	[1.5,-1]	[2,-1]
[-2,-.5]	[-1.5,-.5]	[-1,-.5]	[-.5,-.5]	[0,-.5]	[.5,-.5]	[1,-.5]	[1.5,-.5]	[2,-.5]
[-2,0]	[-1.5,0]	[-1,0]	[-.5,0]	[0,0]	[.5,0]	[1,0]	[1.5,0]	[2,0]
[-2,.5]	[-1.5,.5]	[-1,.5]	[-.5,.5]	[0,.5]	[.5,.5]	[1,.5]	[1.5,.5]	[2,.5]
[-2,1]	[-1.5,1]	[-1,1]	[-.5,1]	[0,1]	[.5,1]	[1,1]	[1.5,1]	[2,1]
[-2,1.5]	[-1.5,1.5]	[-1,1.5]	[-.5,1.5]	[0,1.5]	[.5,1.5]	[1,1.5]	[1.5,1.5]	[2,1.5]
[-2,2]	[-1.5,2]	[-1,2]	[-.5,2]	[0,2]	[.5,2]	[1,2]	[1.5,2]	[2,2]

Why are we choosing displacement values in the range of -2 to 2?

Standard Normal Distribution

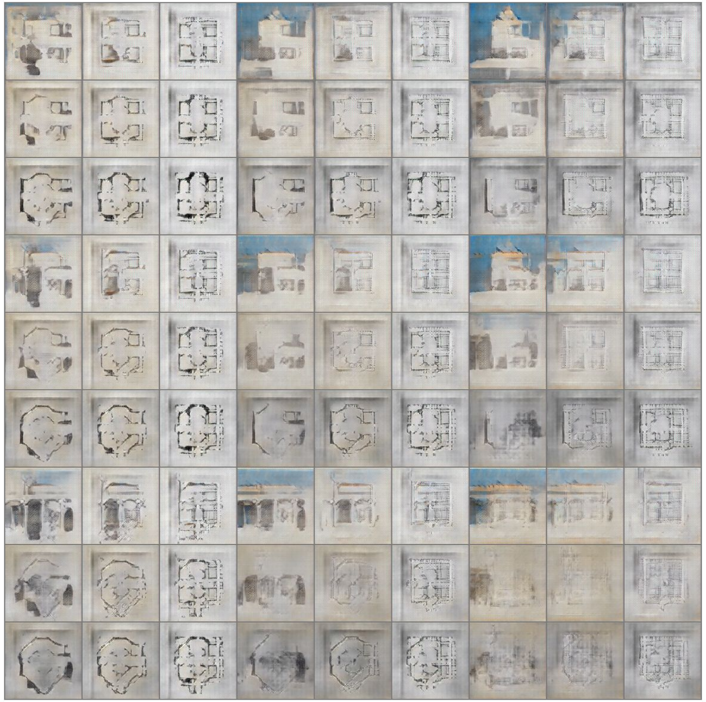


−0.86	0.39	1.67	1.02	−2.22	1.09	−0.99	0.21	0.10	−0.34	−0.28	1.27	2.08	0.90	1.24	0.26	−1.17	−0.71	−1.54	−2.73	0.87	0.79	1.15	1.02	−0.19
1.03	0.57	1.02	−0.42	−0.03	0.79	1.51	0.83	0.05	0.58	−0.72	1.27	0.82	−0.76	−0.09	−2.07	1.13	1.57	0.83	0.06	−1.41	1.45	0.71	0.52	−0.34
−0.41	0.91	0.80	−0.44	0.09	−0.37	−0.99	0.36	−0.66	−0.38	−0.56	0.31	1.22	−1.41	−0.28	0.38	−0.70	−0.88	0.07	0.21	−1.20	0.59	−0.45	−2.68	1.33
0.59	−0.93	0.86	0.79	0.76	0.65	0.93	0.01	1.68	0.28	−2.60	1.73	−1.25	−1.13	1.46	−1.49	−0.43	0.19	−0.37	0.52	−0.63	0.72	−0.38	1.06	0.55

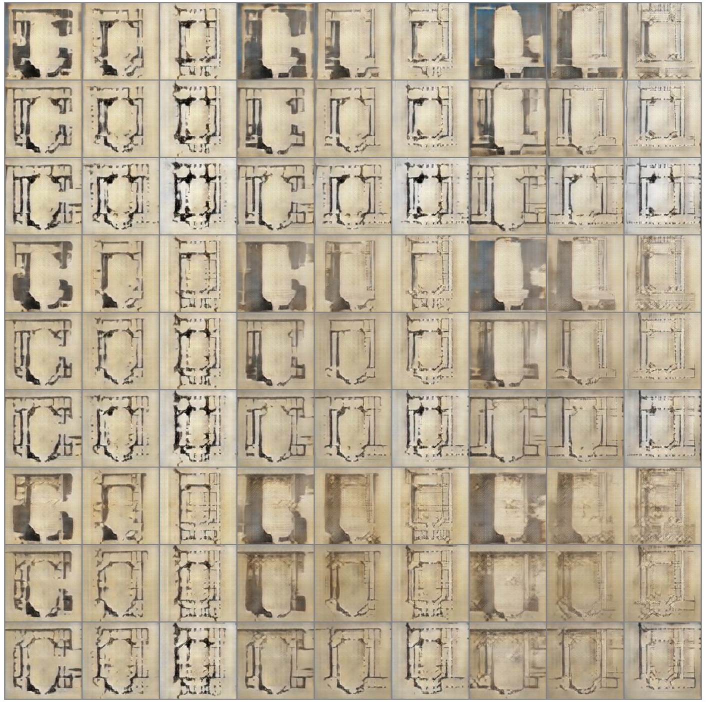
Varied Grid

Varied Grid

Varied Grid



Varied Grid



Varied Grid

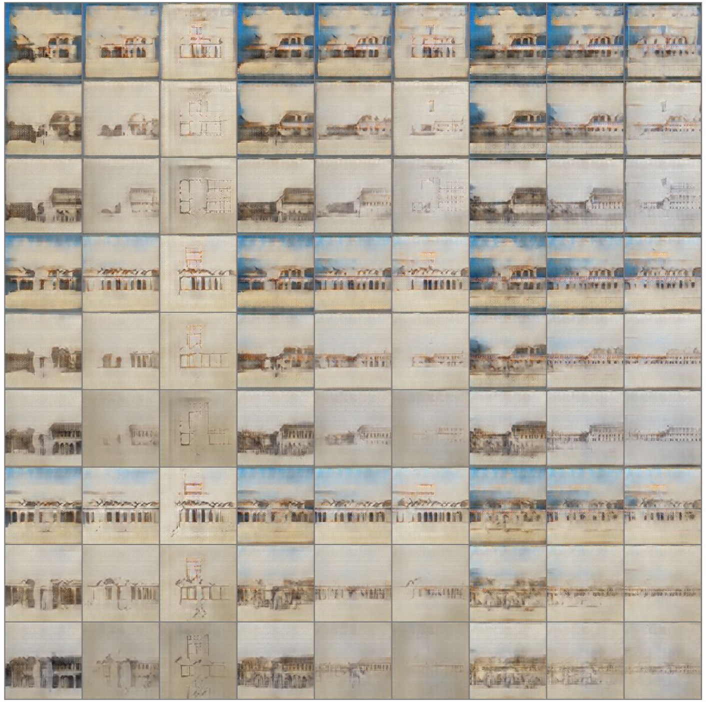
Varied Grid

Varied Grid

The central square is generated from a given 100-dimensional latent vector. It is surrounded by other generated images whose vectors have been displaced a fixed amount from the original latent vector.

The displacement for each dimension is either -1, 0, or 1.

The displacement can be represented by a 100-dimensional vector where each value is -1, 0, or 1.

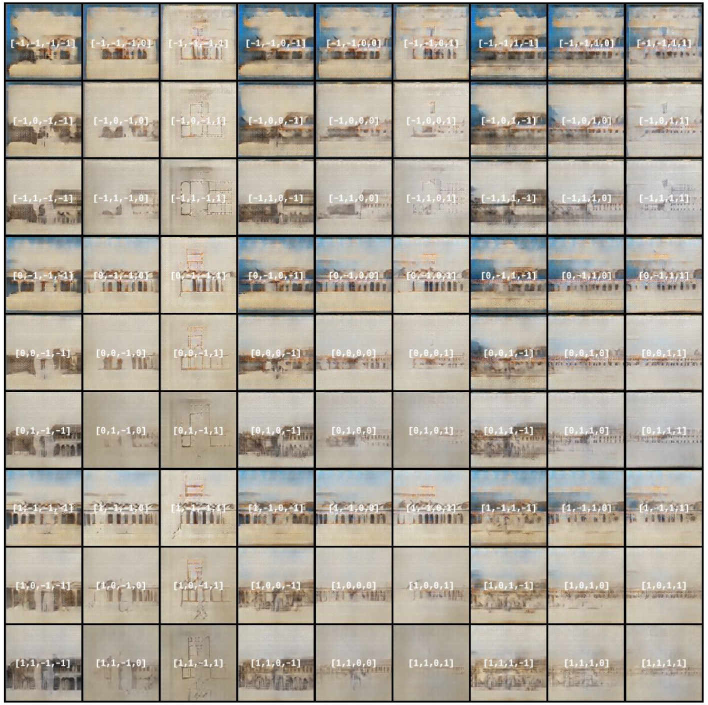


Varied Grid

Reducing the latent displacement vector into a 4-dimensional representation. i.e. the vector below:

[0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,
-1,-1,-1,-1,-1,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0]

is represented as [0,1,-1,0] in the diagram.



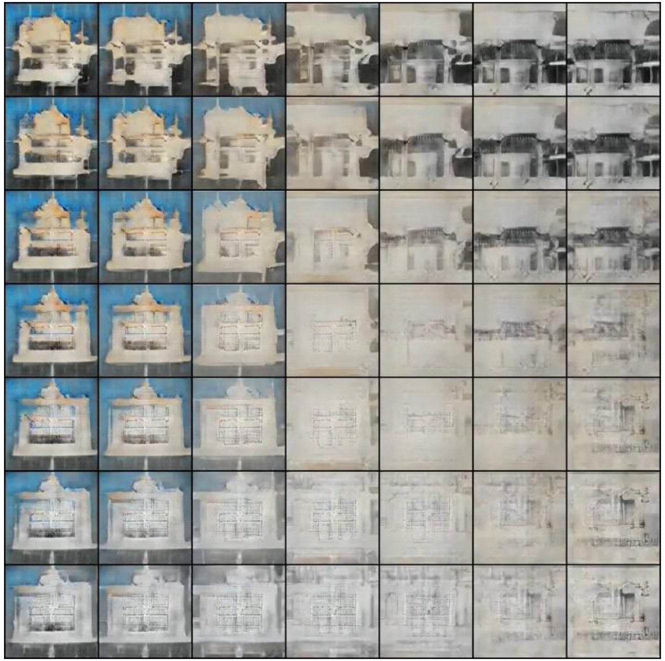
Reducing the latent displacement vector into a 4-dimensional representation. i.e. the vector below:

[0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,
-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,
-1,-1,-1,-1,-1,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0]

is represented as [0,1,-1,0] in the diagram.

[-1,-1,-1,-1]	[-1,-1,-1,0]	[-1,-1,-1,1]	[-1,-1,0,-1]	[-1,-1,0,0]	[-1,-1,0,1]	[-1,-1,1,-1]	[-1,-1,1,0]	[-1,-1,1,1]
[-1,0,-1,-1]	[-1,0,-1,0]	[-1,0,-1,1]	[-1,0,0,-1]	[-1,0,0,0]	[-1,0,0,1]	[-1,0,1,-1]	[-1,0,1,0]	[-1,0,1,1]
[-1,1,-1,-1]	[-1,1,-1,0]	[-1,1,-1,1]	[-1,1,0,-1]	[-1,1,0,0]	[-1,1,0,1]	[-1,1,1,-1]	[-1,1,1,0]	[-1,1,1,1]
[0,-1,-1,-1]	[0,-1,-1,0]	[0,-1,-1,1]	[0,-1,0,-1]	[0,-1,0,0]	[0,-1,0,1]	[0,-1,1,-1]	[0,-1,1,0]	[0,-1,1,1]
[0,0,-1,-1]	[0,0,-1,0]	[0,0,-1,1]	[0,0,0,-1]	[0,0,0,0]	[0,0,0,1]	[0,0,1,-1]	[0,0,1,0]	[0,0,1,1]
[0,1,-1,-1]	[0,1,-1,0]	[0,1,-1,1]	[0,1,0,-1]	[0,1,0,0]	[0,1,0,1]	[0,1,1,-1]	[0,1,1,0]	[0,1,1,1]
[1,-1,-1,-1]	[1,-1,-1,0]	[1,-1,-1,1]	[1,-1,0,-1]	[1,-1,0,0]	[1,-1,0,1]	[1,-1,1,-1]	[1,-1,1,0]	[1,-1,1,1]
[1,0,-1,-1]	[1,0,-1,0]	[1,0,-1,1]	[1,0,0,-1]	[1,0,0,0]	[1,0,0,1]	[1,0,1,-1]	[1,0,1,0]	[1,0,1,1]
[1,1,-1,-1]	[1,1,-1,0]	[1,1,-1,1]	[1,1,0,-1]	[1,1,0,0]	[1,1,0,1]	[1,1,1,-1]	[1,1,1,0]	[1,1,1,1]

Linear Grid Interpolation

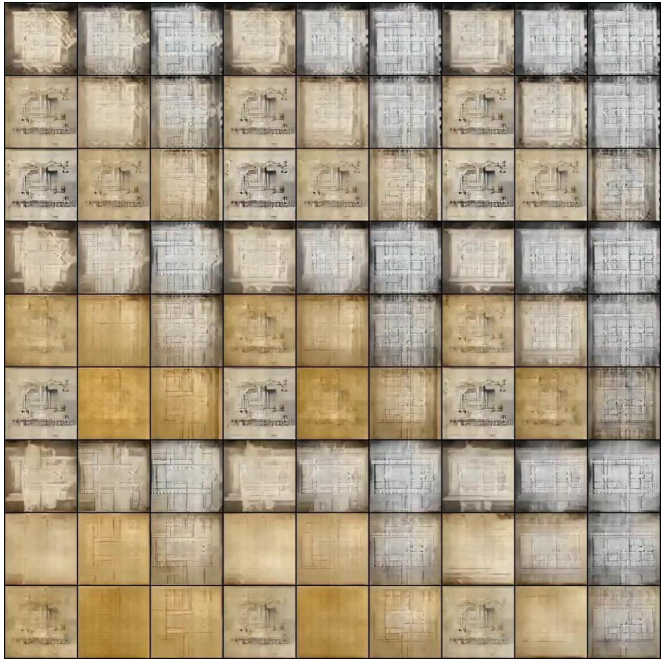


How can we make sure we are exploring/varying all the dimensions in the latent space?

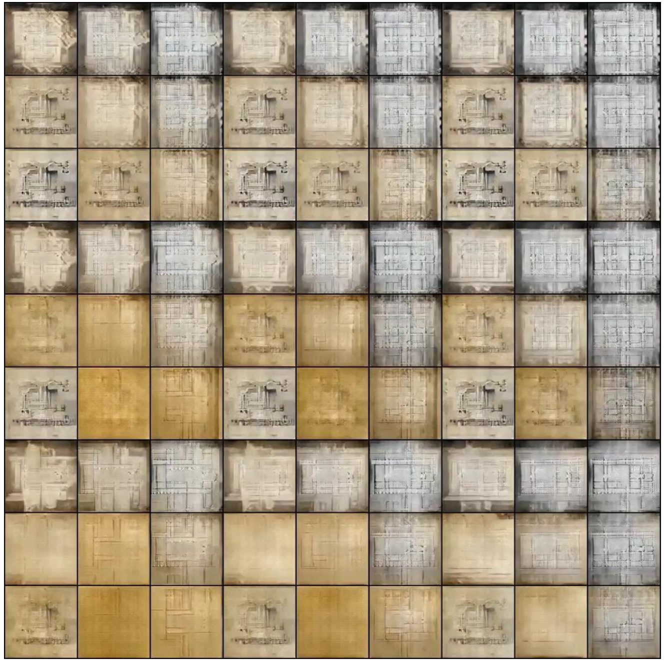
Is there some systematic way to view all results across the space?

Combination/Integer Iteration

Integer Iteration



Integer Iteration



Vector Arithmetic

Vector Arithmetic

$$A - B + C = ?$$

Vector Arithmetic

A	B
?	C

Vector Arithmetic

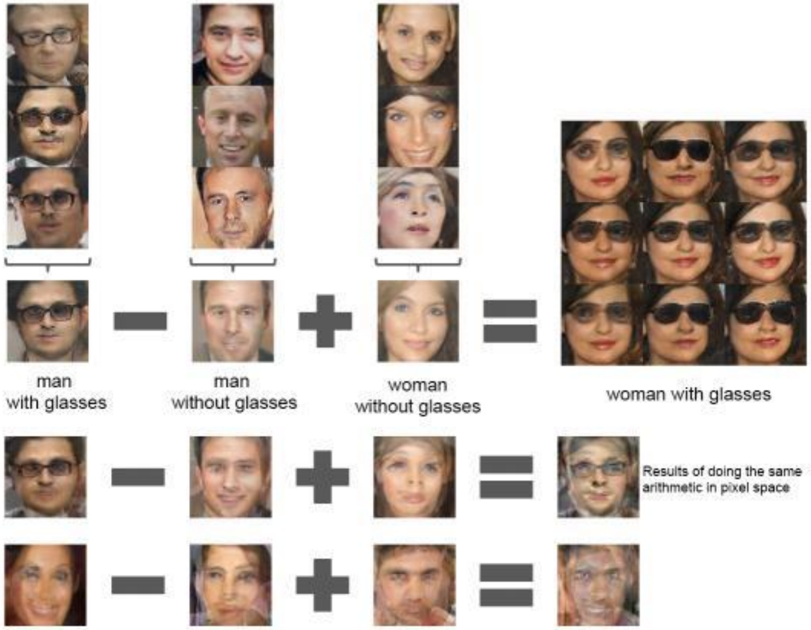
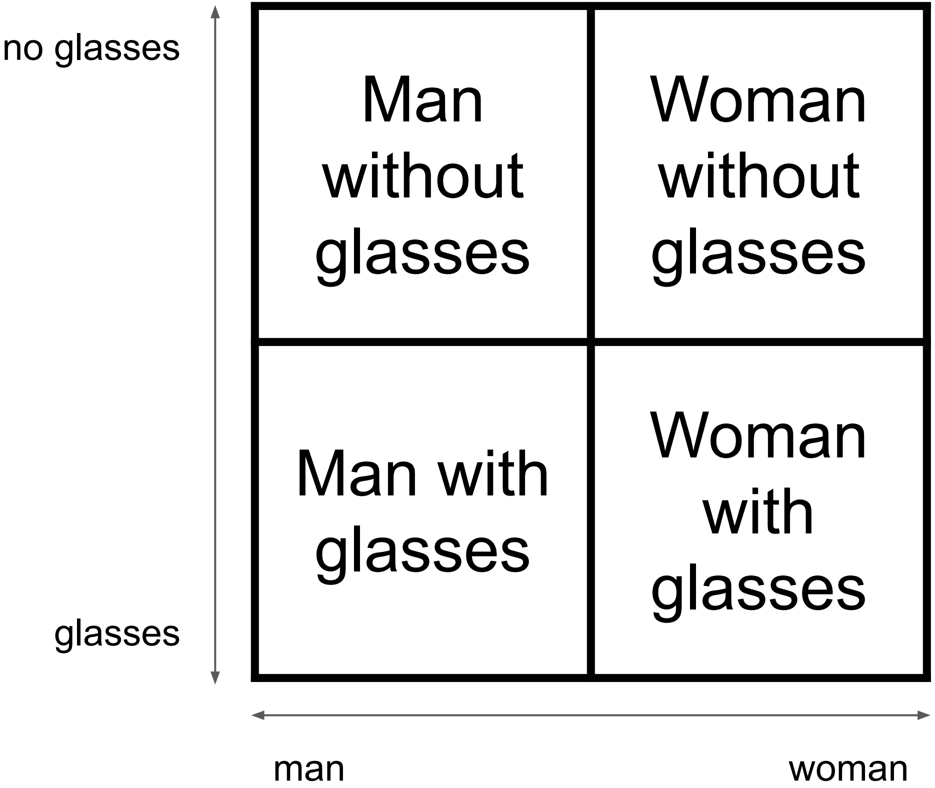


Figure 7: Vector arithmetic for visual concepts. For each column, the Z vectors of samples are averaged. Arithmetic was then performed on the mean vectors creating a new vector Y. The center sample on the right hand side is produce by feeding Y as input to the generator. To demonstrate the interpolation capabilities of the generator, uniform noise sampled with scale +-0.25 was added to Y to produce the 8 other samples. Applying arithmetic in the input space (bottom two examples) results in noisy overlap due to misalignment.

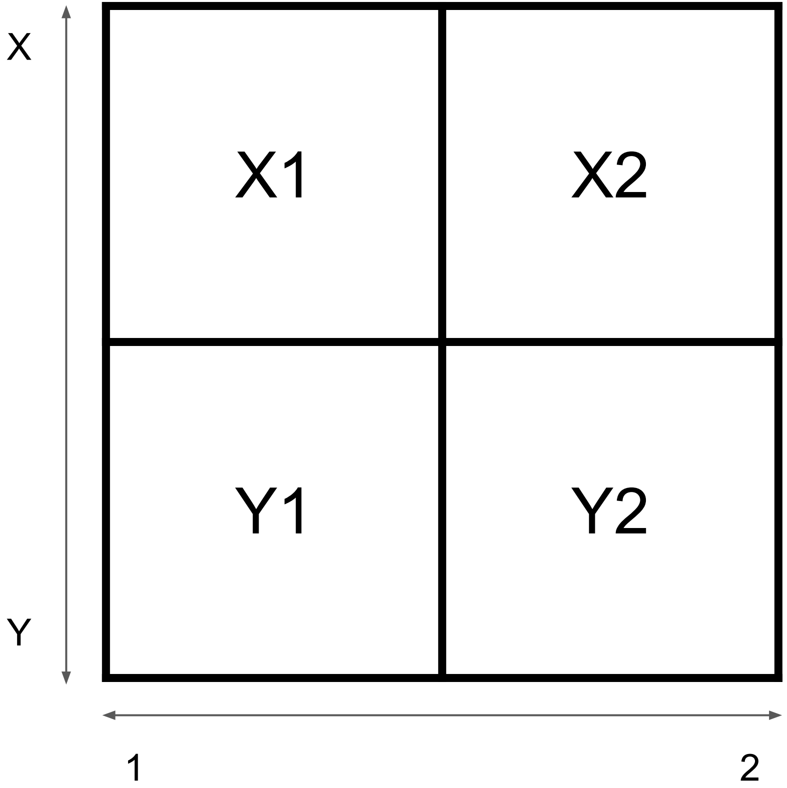
From Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

Vector Arithmetic Step 1: Look for polarized/opposite categories in your results



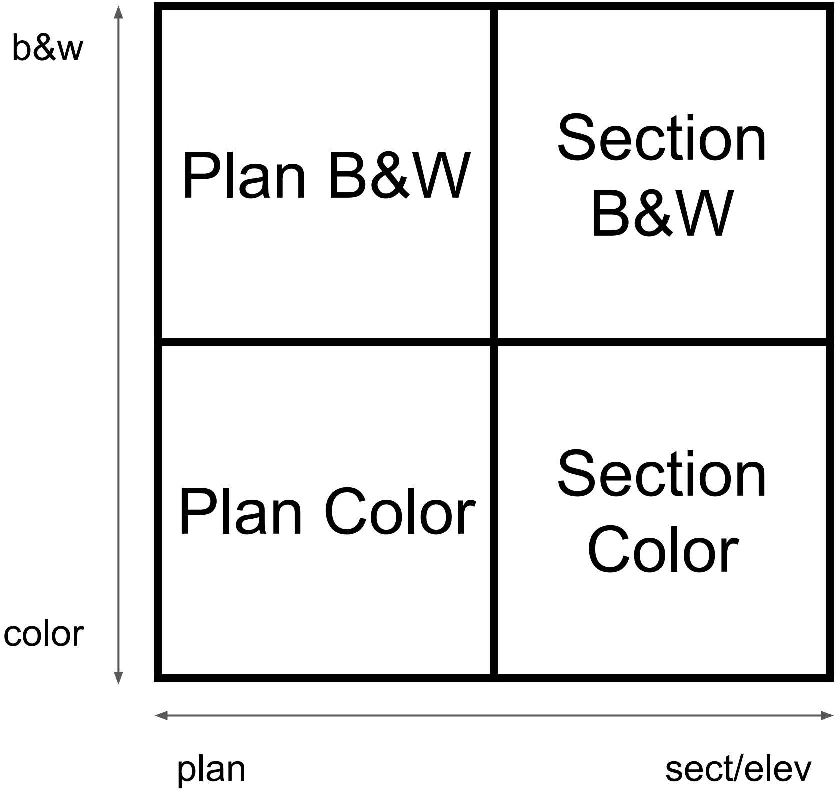
	man	woman
no glasses	Man without glasses	Woman without glasses
glasses	Man with glasses	Woman with glasses

Vector Arithmetic Step 1: Look for polarized/opposite categories in your results



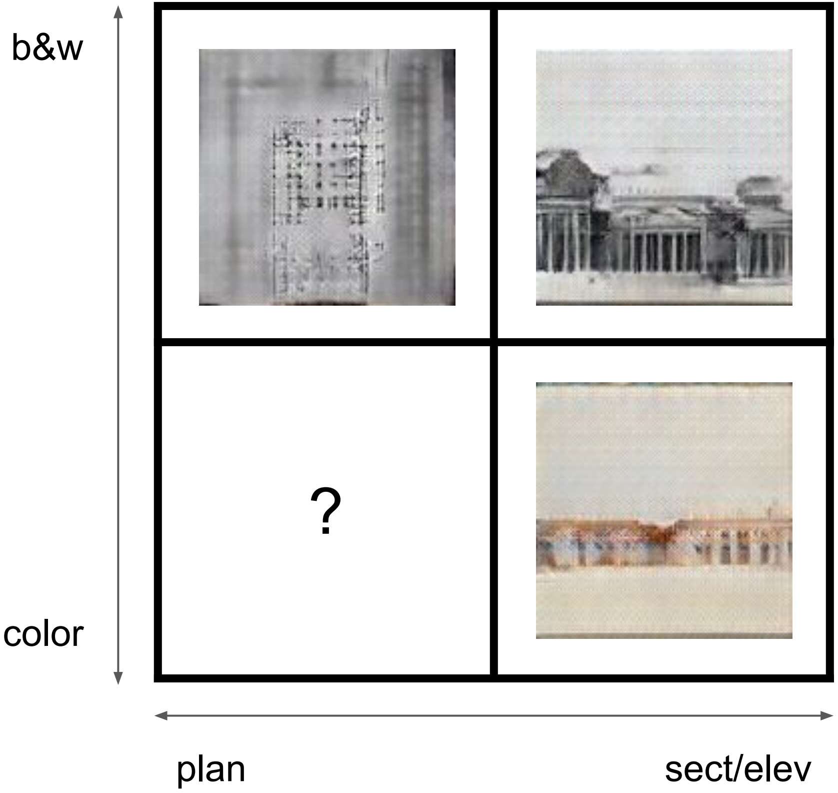
	1	2
X	X1	X2
Y	Y1	Y2

Vector Arithmetic Step 1: Look for polarized/opposite categories in your results



	plan	sect/elev
b&w	Plan B&W	Section B&W
color	Plan Color	Section Color

Vector Arithmetic Step 2: For 3 out of 4 categories, find 3 examples



b&w

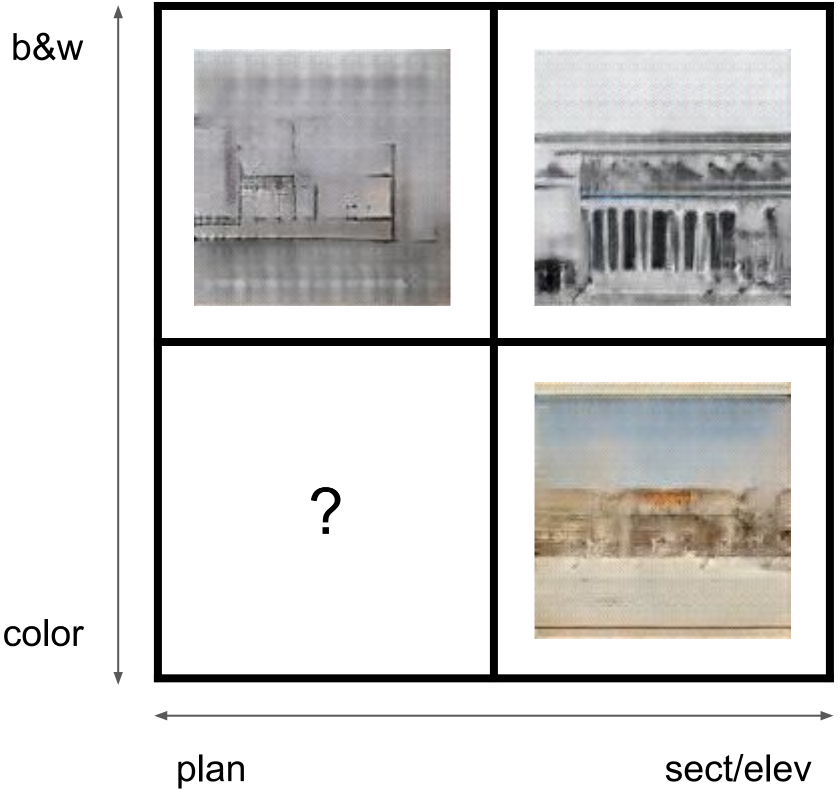
color

?

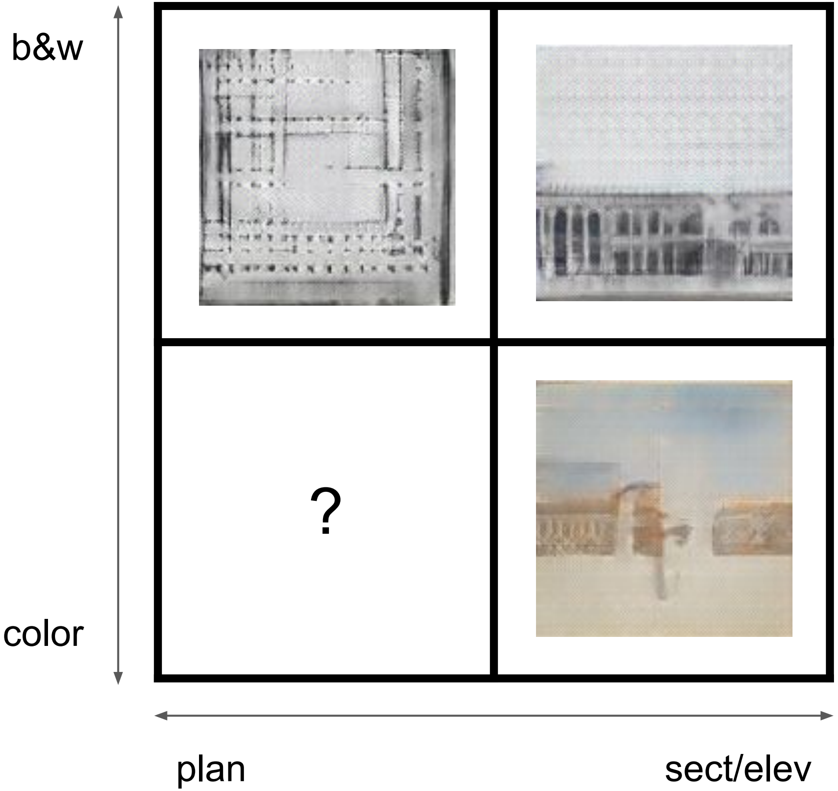
plan

sect/elev

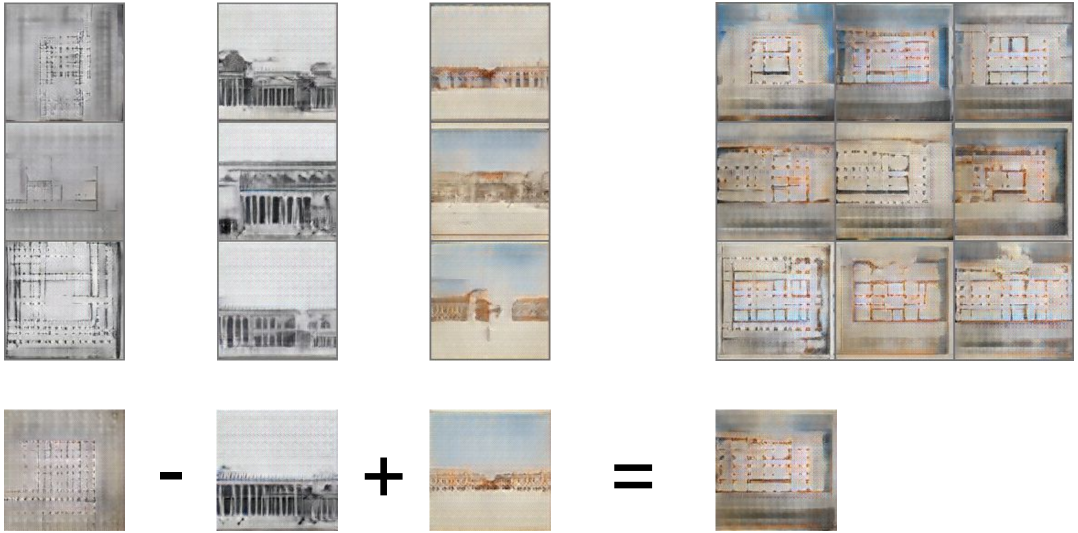
Vector Arithmetic Step 2: For 3 out of 4 categories, find 3 examples



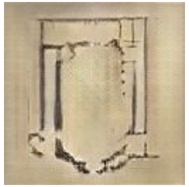
Vector Arithmetic Step 2: For 3 out of 4 categories, find 3 examples



Vector Arithmetic



Curating Individual Latent Vectors (Beaux-Arts Example)

Image					
Random Seed	0	15	73	80	98
Image Index	52	46	33	32	60
Vector Index	0	1	2	3	4

HW 5

Final Project / Presentation on Friday!