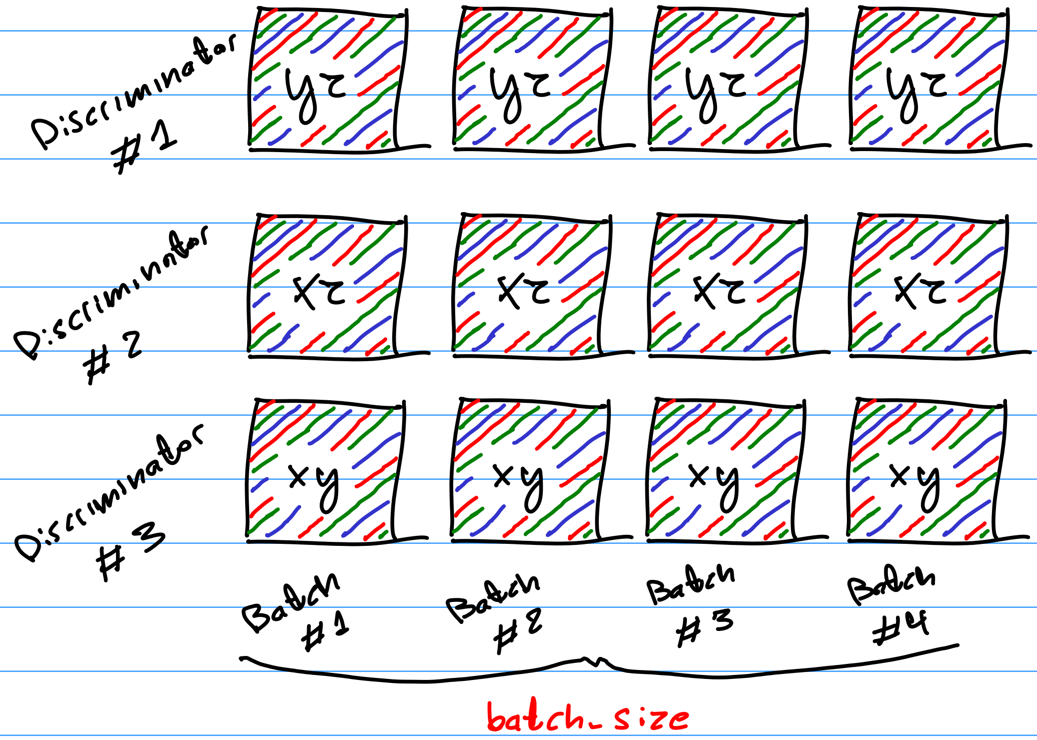
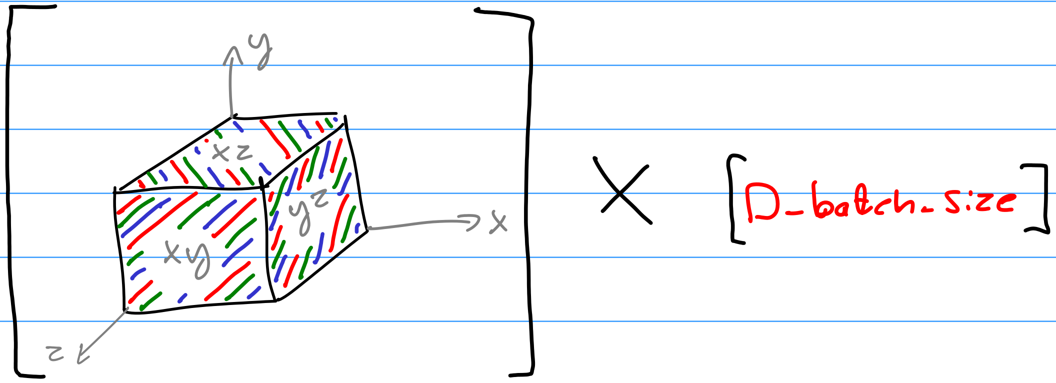
Real



Fake



The Fake data exists as a 5th dimensional tensor:

[D_batch_size, nc, z, y, x]

↳ dims (image size, same in all directions)

↳ number of channels (Quats 1-4, FeatureIds, etc...)

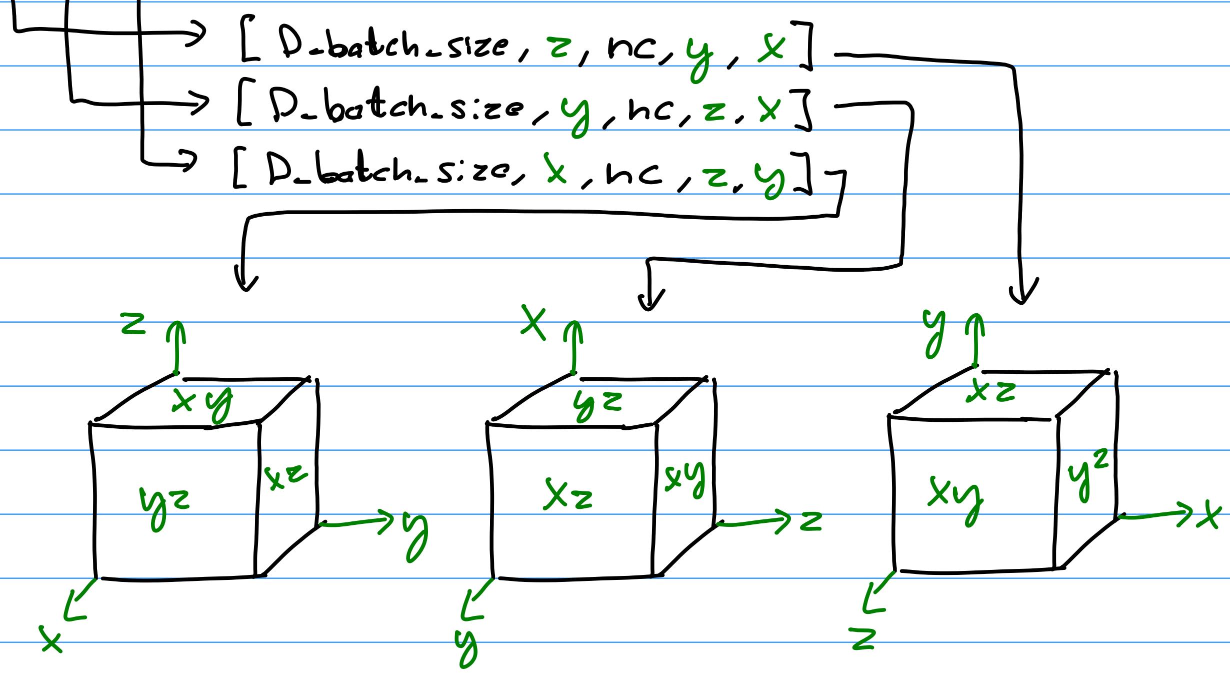
↳ Number of generated 3-D microstructures

To use this 5-D tensor in a discriminator that can only parse 4-D tensors, it must be reshaped...

$$\texttt{data\_fake\_perm = }\underbrace{\texttt{data\_fake.permute(*c\_perm\_dim)}}_{1}\underbrace{\texttt{.reshape(*shape\_disc)}}_{2}$$

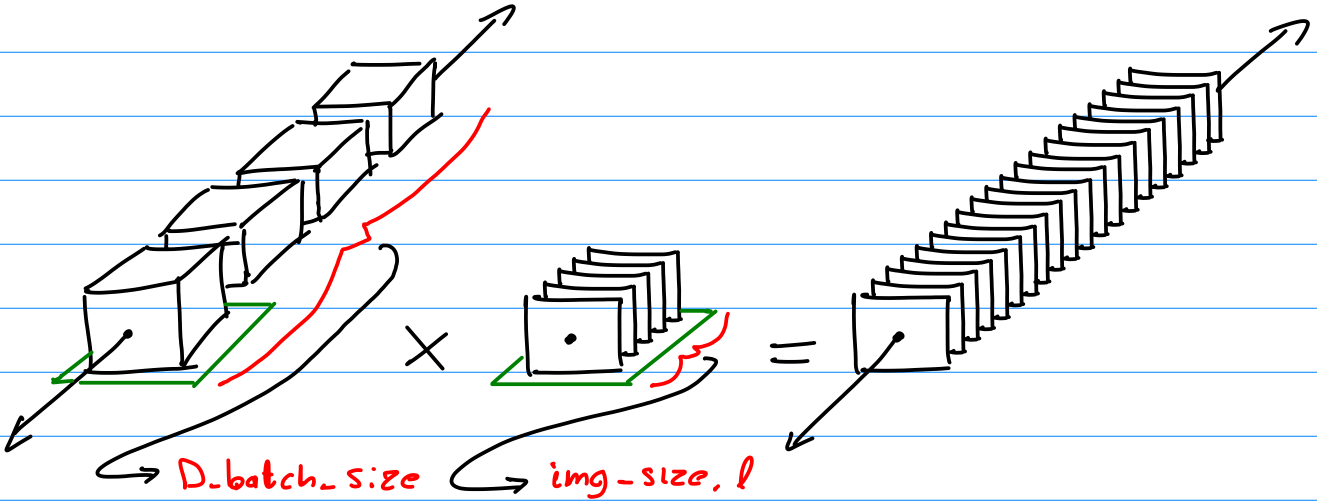
1 Reorganize and Rotate the data cube

```
c_perm = [
    [0, 2, 1, 3, 4], # last 3 indecies are now [nc,y,x] -> z normal
    [0, 3, 1, 2, 4], # last 3 indecies are now [nc,z,x] -> y normal
    [0, 4, 1, 2, 3]  # last 3 indecies are now [nc,z,y] -> x normal
```


[D_batch_size, z, nc, y, x]
[D_batch_size, y, nc, z, x]
[D_batch_size, x, nc, z, y]
z
xy
yz
xz
y
x
x
yz
xz
xy
z
y
y
xz
xy
yz
x
z

2 Reshape the tensor

- shape_disc = [l * D_batch_size, nc, l, l]
  # combine batch size and slice normal plane
  # (turn the entire unused direction into batches)
- X, Y, and Z dimensions all have the same length (image size [l])
- By reshaping l × D_batch_size, sliceGan is lining up the data cubes along the face of interest's normal vector :



Each rotate +permute operation yeilds a large batch of 2-D data as a 4-D tensor

img_size (l) = 64

# Inputs		#outputs		
batch_size	D_batch_size	real	fake	
8	2	8	128	✓
16	4	16	256	✓
64	1	64	64	✓