Presentation structure

Cover title -

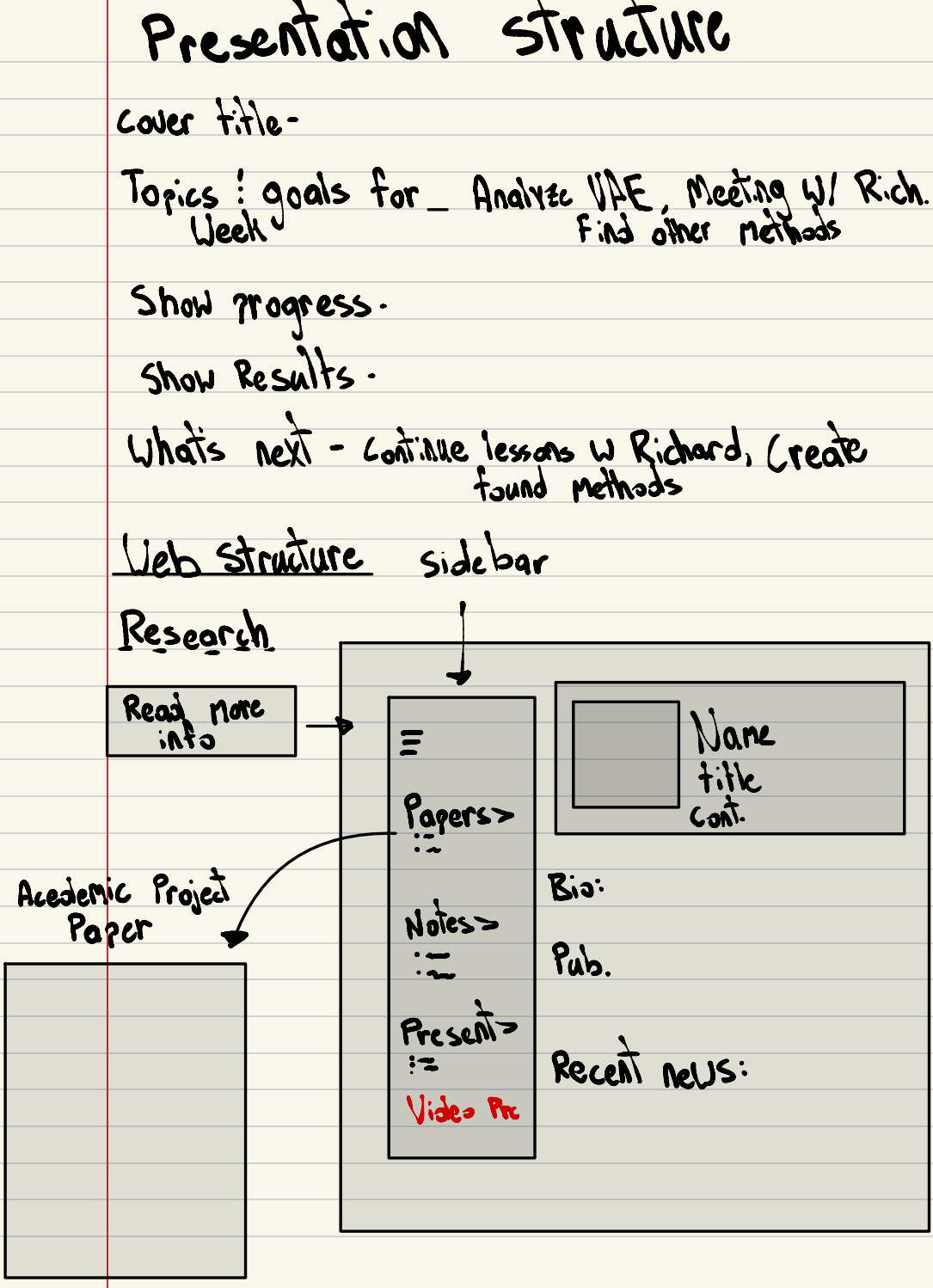
Topics ! goals for _ Analyze VAE, Meeting w/ Rich., Find other methods
Week

Show progress.

Show Results.

Whats next - Continue lessons w Richard, Create found methods

Web Structure



Analyzing PTKFO_VAE.ipynb

256

256

- image loaded using DM3Reader (extracts data)

goals

- Analyze & understand notebook
- Research techniques used for images
- find or build on top of those techniques

```python
def get_image_stacks(path, channel):
    file_list = sorted(os.listdir(path))

    # Initialize lists to hold images of each type and size
    StackFolder_ibw_256 = []


    # Loop through each file in the directory
    for file_name in file_list:
        file_path = os.path.join(path, file_name)

        if file_name.endswith('.ibw'):
            try:
                # Load the image using DM3Reader
                image_data = IgorIBWReader(file_path).read()

                image = np.array(image_data[channel])
                image_shape = image.shape

                # Add the image to the corresponding stack based on its dimensions
                if image_shape == (256, 256):
                    StackFolder_ibw_256.append(image)
                else:
                    print(f"Image '{file_name}' skipped due to unexpected size {image_shape}")

            except Exception as e:
                print(f"Error reading .dm3 file '{file_name}': {str(e)}")

    # Convert the image lists to NumPy arrays
    StackFolder_ibw_256 = np.array(StackFolder_ibw_256)

    return StackFolder_ibw_256
```

```python
def extract_subimages(image, step_size, window_size):
    subimages_target = []
    coms_target = []

    img_height, img_width = image.shape
    window_height, window_width = window_size

    for y in range(0, img_height - window_height + 1, step_size):
        for x in range(0, img_width - window_width + 1, step_size):
            # Extract subimage
            subimage = image[y:y + window_height, x:x + window_width]
            subimages_target.append(subimage)

            # Calculate the center of mass for the subimage
            center_y = y + window_height // 2
            center_x = x + window_width // 2
```

What is the code doing?

get_image_stacks(path, channel)

- lists all the files in path || for each file ending .ibw, it tries to read it with Igor IBW Reader || it grabs 1 channel from (image-data[channel])
- Returns a Numpy array with shape (N, 256, 256) where N is the # of valid images

Notes: .ibw is IGOR Pro's "binary wave" format.

Analyzing PTKFO_VAE.ipynb

techniques used

- VAE = Variational Autoencoder
- latent space with KDE

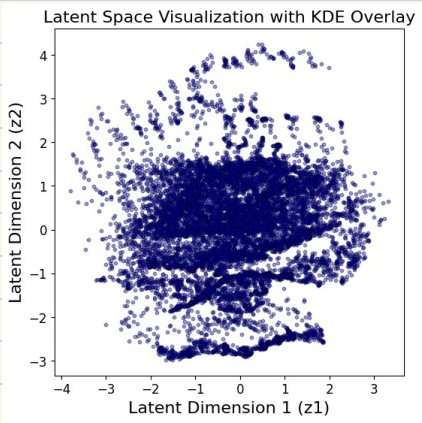
latent space with KDE

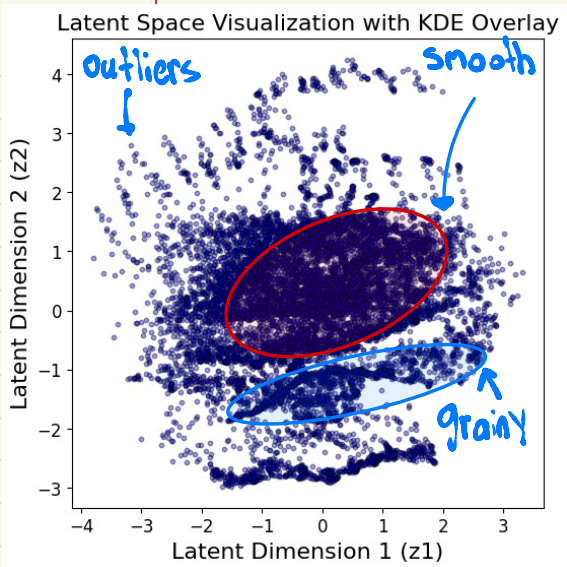
latent space visualization with KDE overlay is a technique used in ML to understand / interpret the structure of data processed by a neural network

- it combines a reduced-dem plot of the data's latent space with a KDE to highlight areas of higher data conc.

1 dot = One image patch (a small cut of the image)

- it shows patches that look similar are close together, patches that look different are far apart

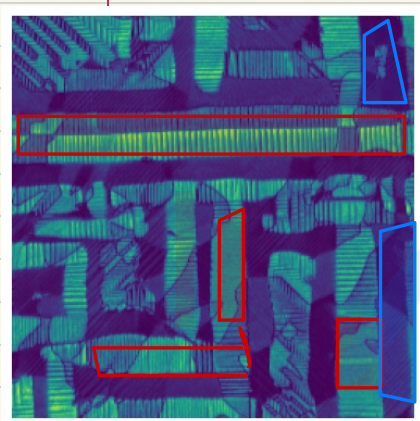




Why do dots cluster together?

- Some dots show smooth flat areas
- Some show grainy textures
- some show lines or cracks

- All the "smooth" patches end up with similar z1, z2 values → form a cluster
- All the "grainy" patches end up in another region → another clust



How to interpret this?

looking at the plot:

Big dense blob in middle: the "typical" structure in dataset (most common mineral/phase in background)

Scattered dots at edges: rare patches might correspond to unusual material features, impurities, or even noise

Why is this important?

- you have a map of all patch types in dataset
- you can project those classes back onto the original image: see where they occur spatially

how can we build on top of this?

- clustering (e.g. k-means)

- why? scatterplot is visually messy. clustering algorithms can automatically groups into "texture classes"
  - you might discover groups that carresponds to a distinct texture type

Condo libraries
to combiterial !!