**Sigma\_map\_generation Version:**  
**# 5th version update: choose the highest peak in whole list, and then choose the peak from [10:] list**

threshold\_high = 0.5

threshold\_high\_map = np.full((height, width), threshold\_high)

# Find maximum values and second maximum values

wd\_sigma\_list\_temp = wd\_sigma\_list

first\_peak\_values = np.max(wd\_sigma\_list\_temp, axis=0)

second\_peak\_values = np.max(wd\_sigma\_list\_temp[10:, :, :], axis=0)

# Compute ratios

ratios = wd\_sigma\_list[5:] / np.maximum(second\_peak\_values, first\_peak\_values)

sigma\_sequence\_final = sigma\_sequence[5:]

# Compute sigma map for threshold\_high

indices\_high = np.argmax(ratios >= threshold\_high\_map, axis=0)

# Exclude the maximum values to find the second highest ratio if needed

max\_values = np.max(ratios, axis=0)

masked\_ratios = np.where(ratios == max\_values, -np.inf, ratios)

second\_indices\_high = np.argmax(masked\_ratios, axis=0)

# Use second highest ratio if the highest ratio is exactly 1

ratios = np.transpose(ratios,[1,2,0])

values = np.take(ratios,indices\_high)

final\_indices\_high = np.where(values >= 1, second\_indices\_high, indices\_high)

sigma\_map\_high = np.take(sigma\_sequence\_final, final\_indices\_high)

# Determine final sigma map

sigma\_map = np.where(first\_peak\_values > 8 \* second\_peak\_values, 0, sigma\_map\_high)