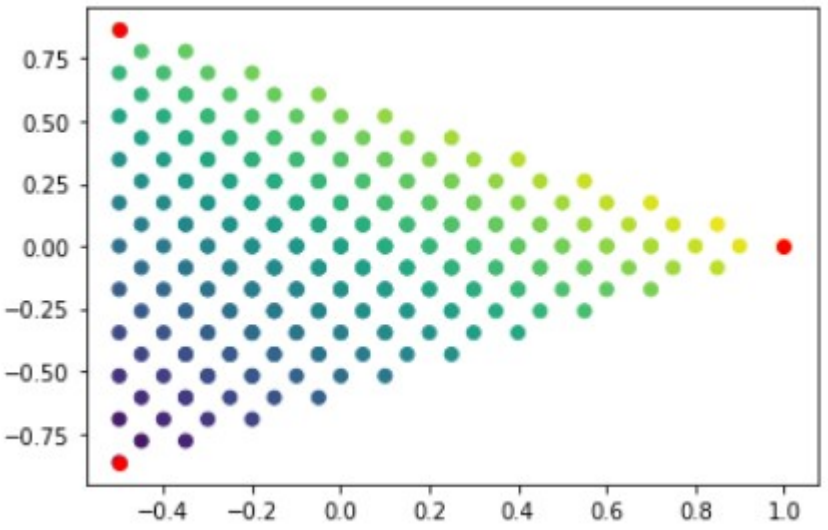
# Data creation

- use points uniformly distributed in 2D ‚base-latent-space‘ with target = x+y
- apply any kind of function to x and y to create coords in real space

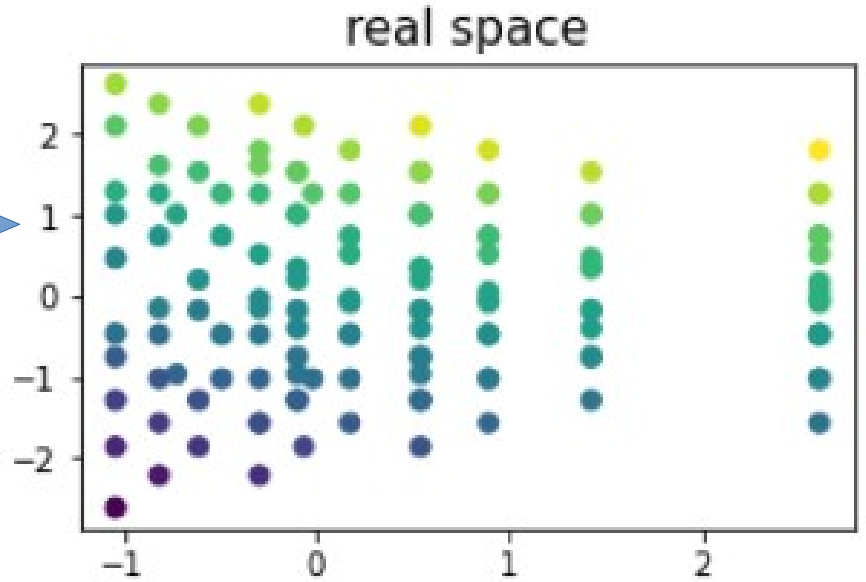
base-latent-space




X‘ = xfunc(x)

Y‘ = yfunc(y)

real space



# Implications of data creation

- like this, for a perfectly working NN the derived latent space should be distributed uniformly, too
- General usefullness of NN can be assessed by
  - comparing distribution in derived latent space to uniform distribution
  - MSE for features and target between real and reconstructed real space

  for transformation functions of varying complexity:
  - 0: no trafo; 1: lin. Trafo; 2: mon. Non-lin. (exp, ^3,^5,…),
  - 3: non-mon. non-linear (Gaussian); 4: periodic (cos,sin)

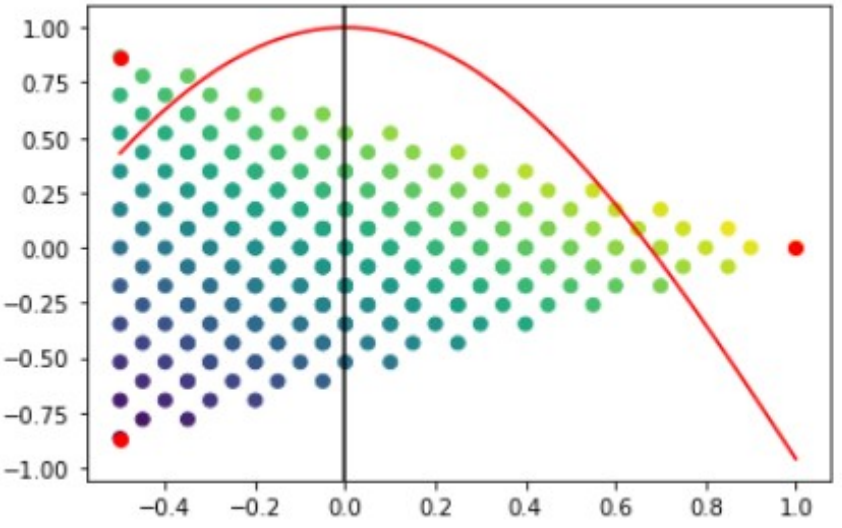
Tuple entries in hyperpar_df stand for (mean,std.dev) respectively:

| | MSE_feats | MSE_targets | lat_gauss | lat_uniform | tot | err |
|---|---|---|---|---|---|---|
| 0 | (0.009, 0.003) | (0.063, 0.211) | (0.558, 0.106) | (0.575, 0.103) | 20 | 1 |
| 1 | (0.011, 0.005) | (0.014, 0.004) | (0.484, 0.105) | (0.502, 0.1) | 20 | 0 |
| 2 | (0.057, 0.093) | (0.445, 0.007) | (0.477, 0.17) | (0.494, 0.16) | 18 | 0 |
| 3 | (0.934, 0.063) | (0.951, 0.211) | (0.52, 0.208) | (0.519, 0.207) | 2 | 0 |
| 4 | (0.027, 0.014) | (0.232, 0.019) | (0.429, 0.072) | (0.448, 0.07) | 16 | 0 |

(all results can be seen in daa_exe)

# Do new points carry additional info? Proof of concept

- x‘ = cos(1.55*x), y‘ = y
- Compare e.g. two cases:
  - Random division: NN trained with rand1, tested with rand2
  - Division at x=0: NN trained with divX<0, tested with divX>0
- Former should work, latter should not




divX<0 divX>0

# Results for Proof of Concept

- how much latent space distribution deviates from uniform distribution is calculated in lat_uniform with Hellinger Distance, in [0,1] → turns 0 for perfect uniform distribution

| | MSE_feats | MSE_targets | lat_uniform |
|---|---|---|---|
| div1div1 | (0.051, 0.067) | (0.02, 0.017) | (0.481, 0.137) |
| div1div2 | (0.043, 0.019) | (2.411, 0.162) | (0.548, 0.051) |
| rand1rand2 | (0.047, 0.036) | (0.239, 0.021) | (0.389, 0.073) |
| div1both | (0.052, 0.045) | (1.038, 0.089) | (0.445, 0.085) |
| bothdiv1 | (0.029, 0.013) | (0.282, 0.2) | (0.507, 0.109) |

Train set

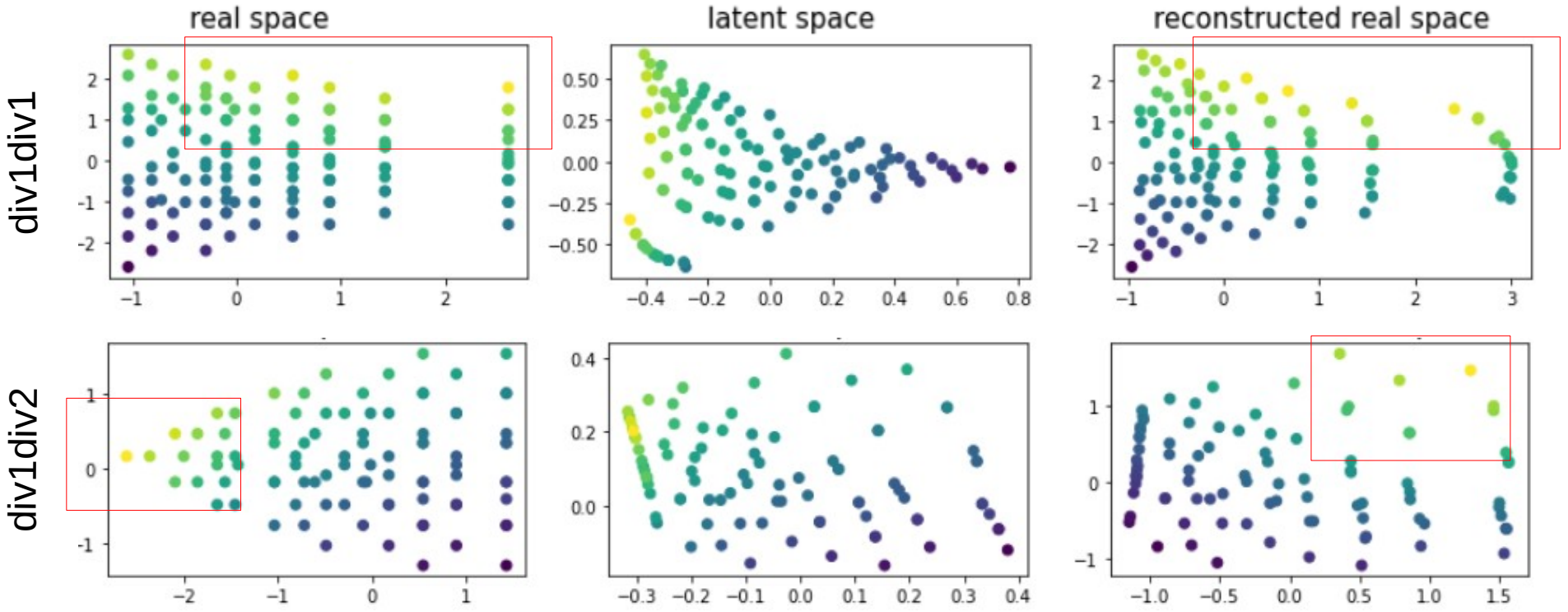
Test set

(mean, std.dev.) for 20 calculations, with each #epochs=1000

- results: MSE_targets seems to be a reliable metric: it increases when new data points carry new information and cannot be predicted well with old model.

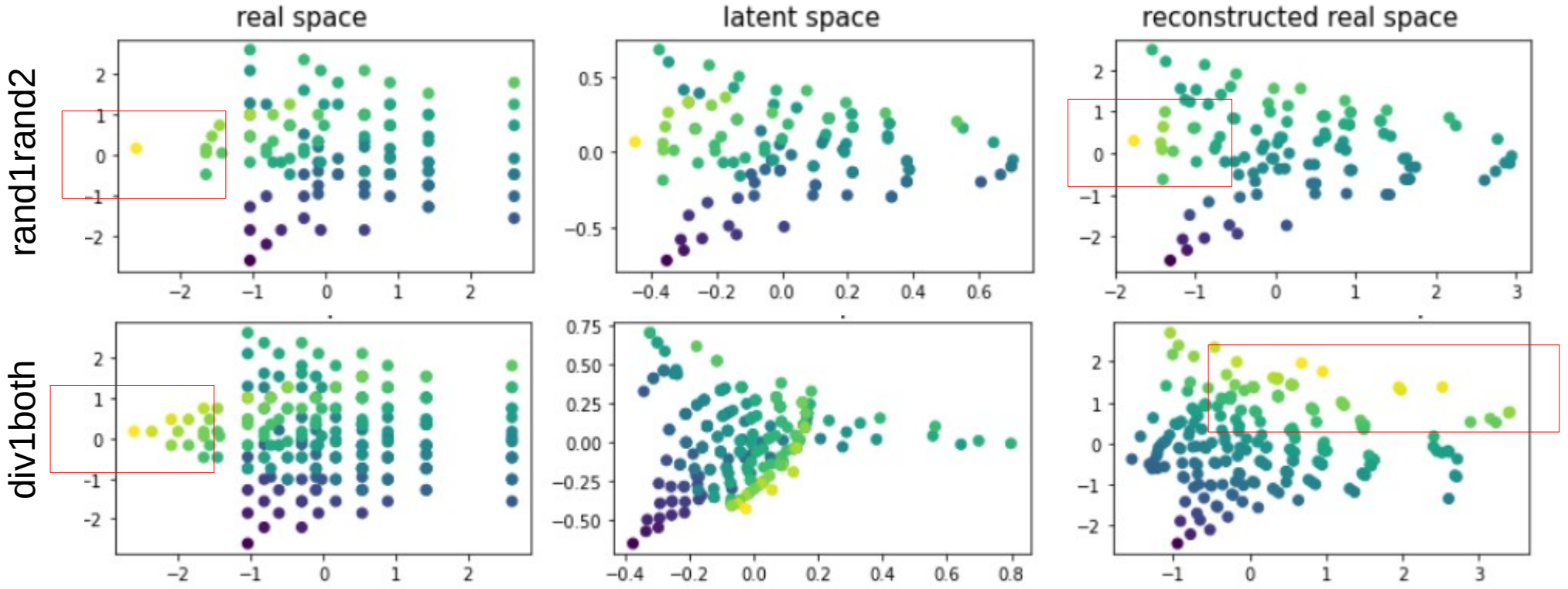
# Results for Proof of Concept

Example calculations: Test points shown in all diagrams



# Results for Proof of Concept

Example calculations: Test points shown in all diagrams




(all results can be seen in crucial_points)

# Result analysis & next steps

- As was to be expected, data points that carry new information cannot be predicted well with old NN
- New points are not displayed outside latent space, hypothesis: setup of NN does not allow for any point to be placed outside
- If target values are unknown for new points, until now not clear if NN suitable for their prediction
- Next step: Check if clustering of new-information carrying points occurs inside latent space