A MULTIOBJECTIVE VIEW ON CREATING COUNTERFACTUAL EXPLANATIONS FOR EXPLAINING UNCERTAINTY IN MACHINE LEARNING

EUROPT 2022, 30th July 2022

Julia Niebling

German Aerospace Centre, Institute of Data Science, Jena, Germany

DLR

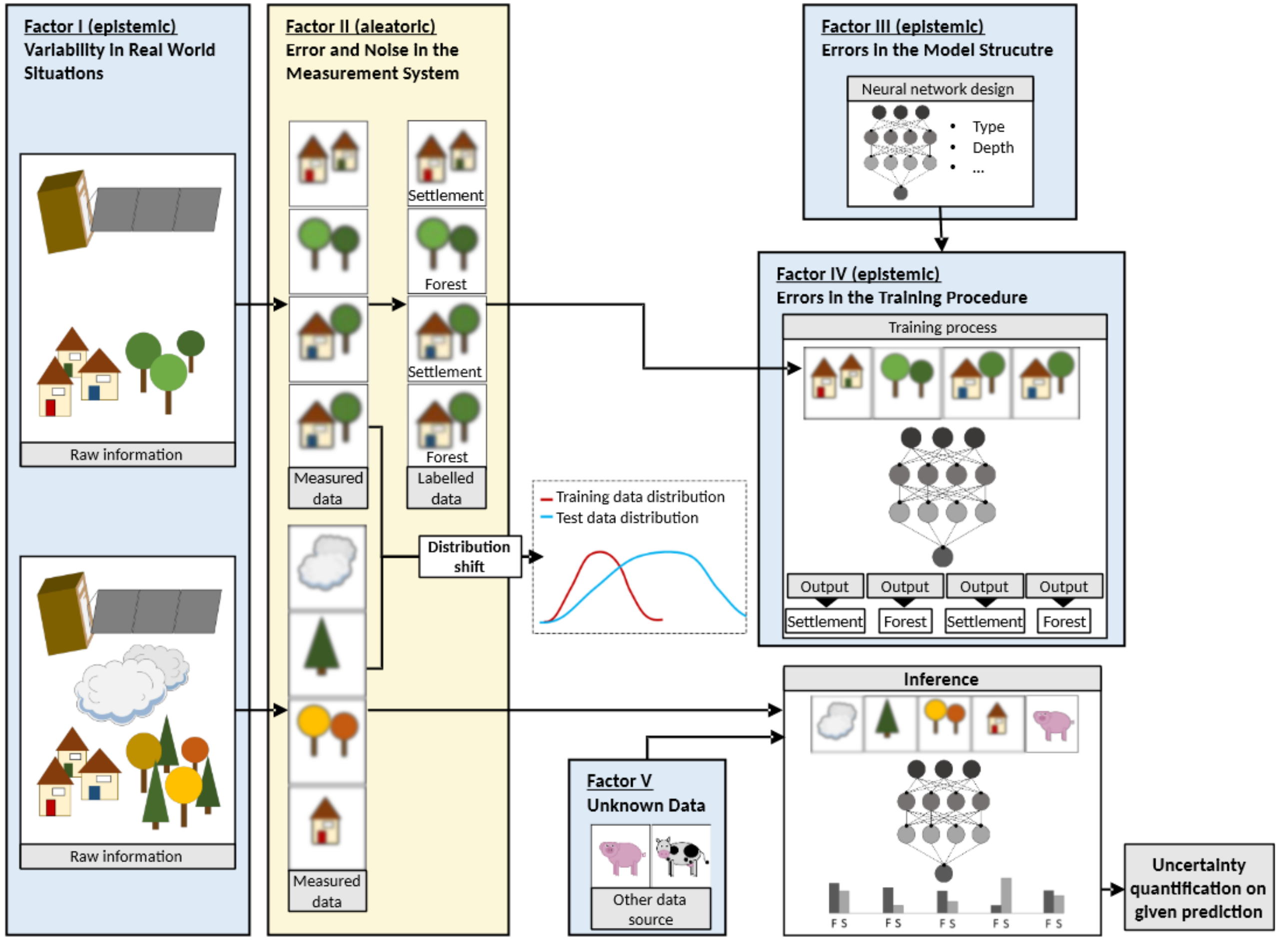
Uncertainty in Machine Learning

- Quantification of uncertainty to
  - Understand models
  - Trust predictions
  - Develop robust methods

Aleatoric (data) uncertainty

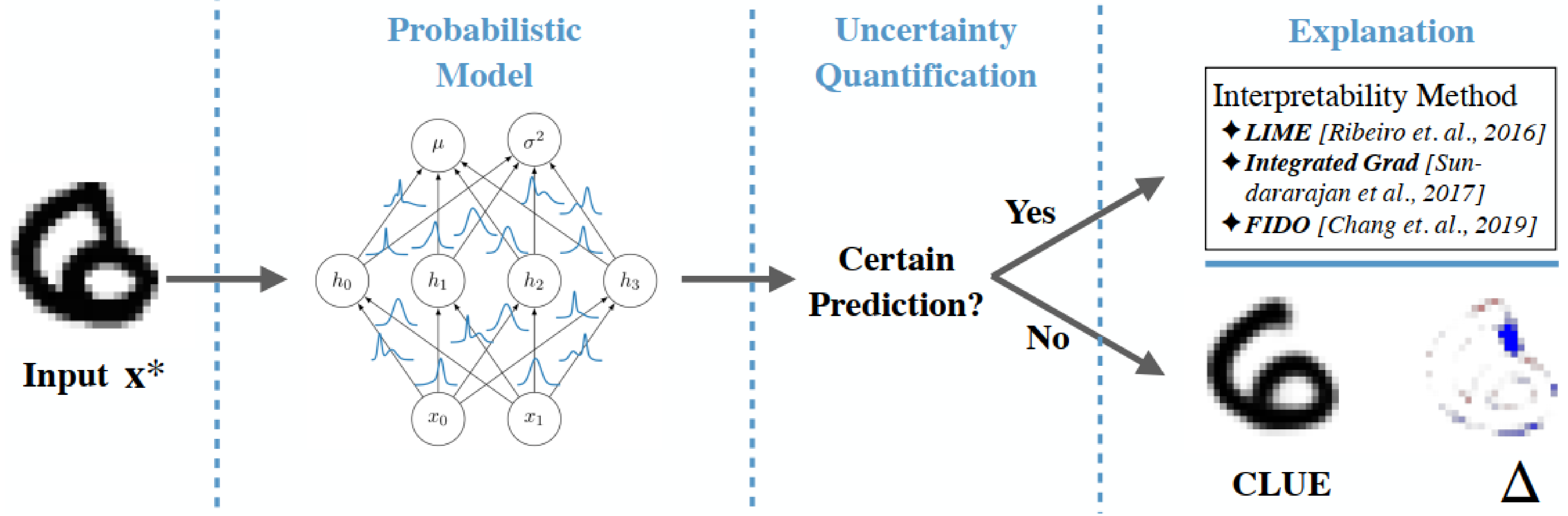
⇕

Epistemic (model) uncertainty



Gawlikowski et.al. A Survey of Uncertainty in Deep Neural Networks

Getting a CLUE



CLUE = *Counterfactual Latent Uncertainty Explanations*

Counterfactual Explanation = *what should be different to change the outcome*

Antorán et.al. Getting a CLUE: A Method for Explaining Uncertainty Estimates (ICLR 2021)

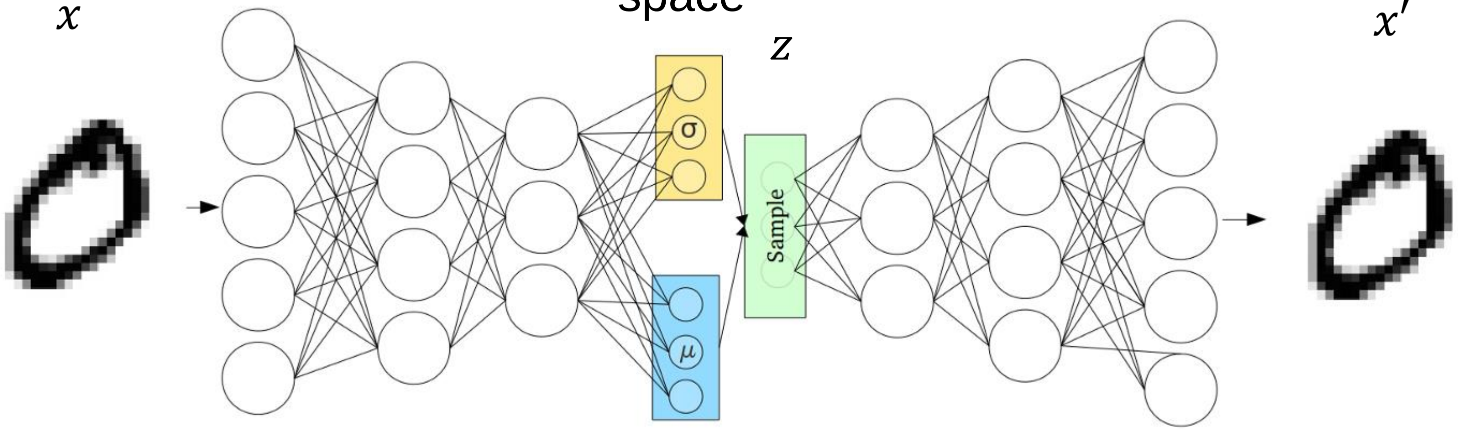
Paraphrasing Optimization ⇔ Machine Learning

- Optimization is used in ML to train models, e.g. neural networks
- Further usage of optimization in ML possible

OPTIMIZATION	MACHINE LEARNING
objective function	loss function
to optimize	to train
iteration (e.g. of gradient descent)	epoch
$x \in \mathbb{R}^n$ (variables in search space)	e.g. $\theta, w \in \mathbb{R}^n$ (parameters of a NN when *training a NN*)

Notations

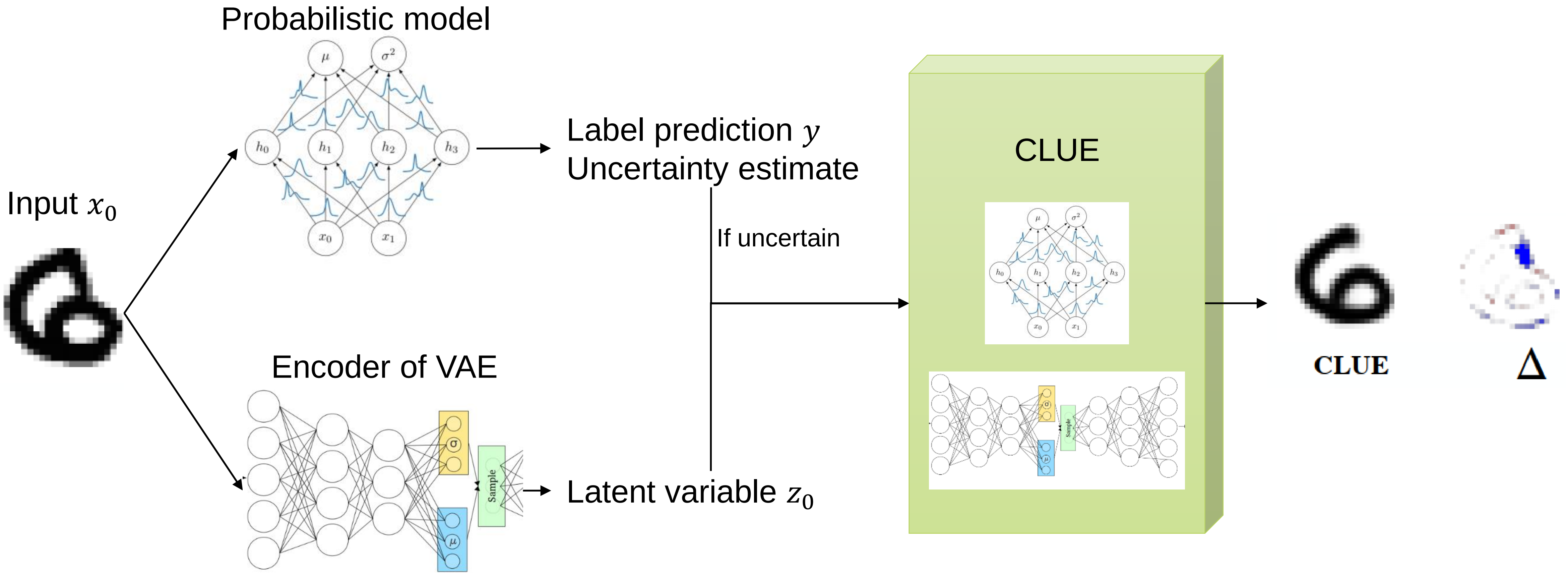
- data points x, labels y, latent space variables z
- probabilistic model $\mathcal{H}$, differentiable, to estimate uncertainty of an observation, e.g. Bayesian NN
- variational autoencoder (VAE) consisting of encoder $q_\Phi(x|z)$ and decoder $p_\Theta(x|z)$ with parameters/weights Φ, Θ



Cf. https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

- predictive means of decoder and encoder of VAE: $E_{p_\Theta(x|z)}[x] =: \mu_\Theta(x|z)$, $E_{q_\Phi(z|x)}[z] =: \mu_\Phi(z|x)$

Pipeline



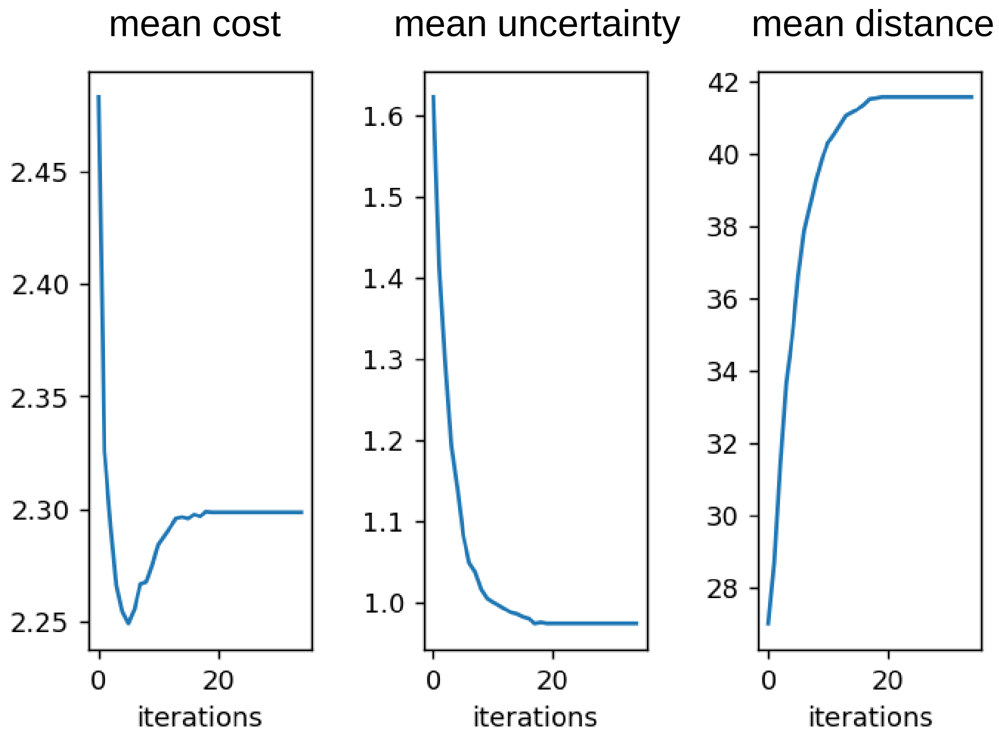
Probabilistic model
Label prediction y
Uncertainty estimate
Input x_0
If uncertain
CLUE
Encoder of VAE
Sample
Latent variable z_0
CLUE
Δ

The optimization problem in CLUE

$$\mathcal{L}(z) = \mathcal{H}(y|\mu_\Theta(x|z)) + d(\mu_\Theta(x|z), x_0)$$

Uncertainty measure
- Stochastic
- Given by trained probabilistic model (e.g. BNN)

Distance metric
- Nearly deterministic, convex
- In sample space, latent space or prediction space



- Solver: Adam, with 35 iterations

Kingma and Ba. Adam: A method for stochastic optimization. (2014)

The multiobjective optimization problem in CLUE

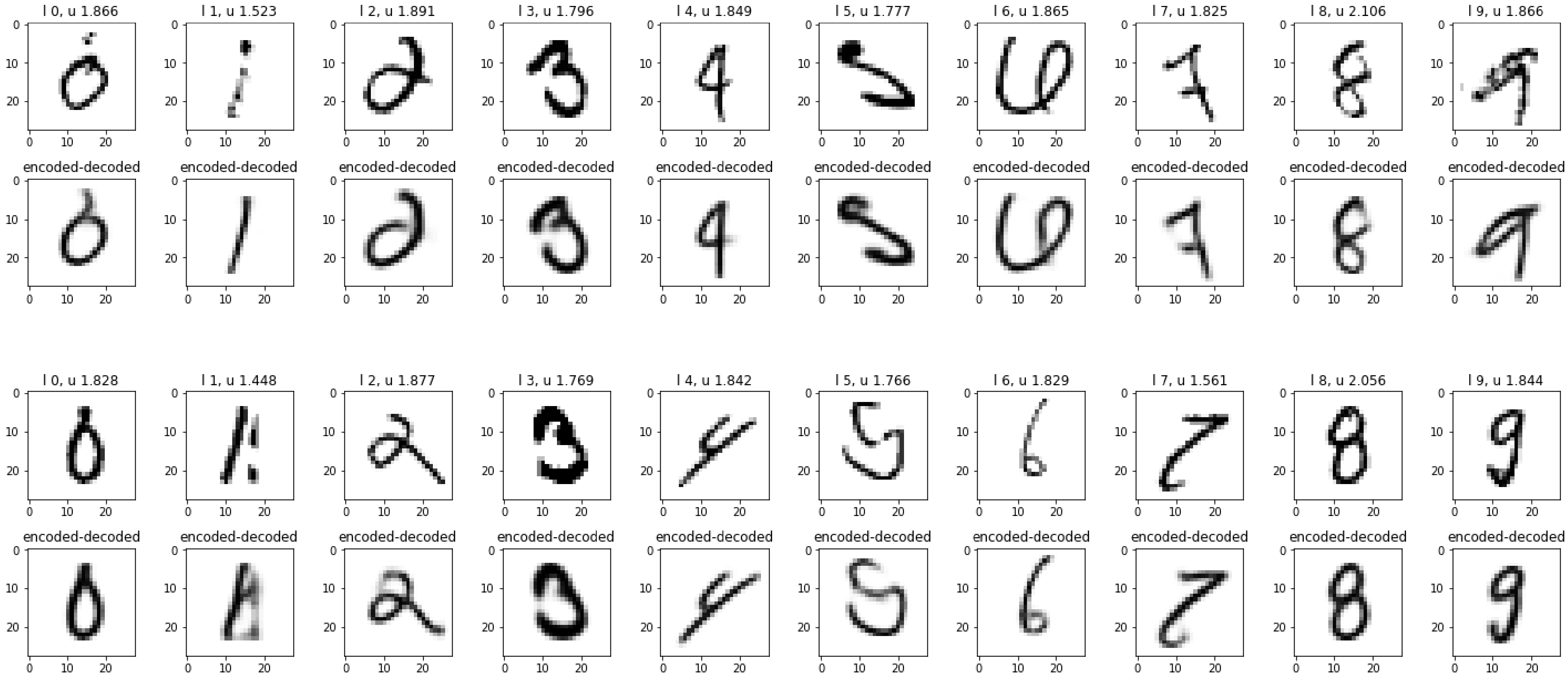
$$\mathcal{L}(z) = \mathcal{H}(y|\mu_\Theta(x|z)) + d(\mu_\Theta(x|z), x_0) \quad \Rightarrow \quad \mathcal{L}(z) = \begin{pmatrix} \mathcal{H}(y|\mu_\Theta(x|z)) \\ d(\mu_\Theta(x|z), x_0) \end{pmatrix}$$

- Two objectives summed up with weight 1
- Different weights to
  - Indicate preferences of decision maker
  - Give more insights to possible solutions
  - …?

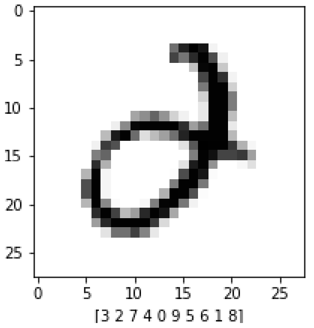
- Solving with weighted sum approach

$$\mathcal{L}(z) = \lambda \cdot \mathcal{H}(y|\mu_\Theta(x|z)) + (1-\lambda) \cdot d(\mu_\Theta(x|z), x_0),$$
$$\lambda \in \{0, 0.1, 0.2, \dots, 1\}$$

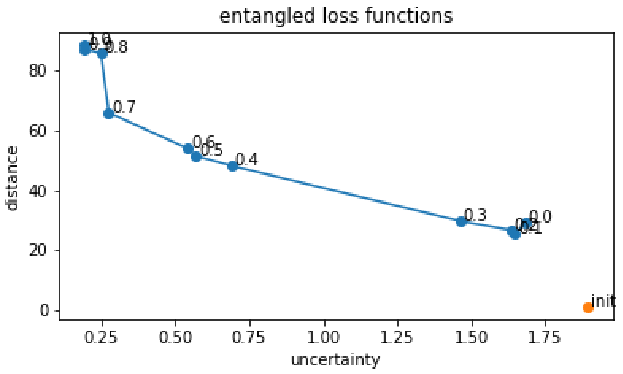
Considered uncertain test samples



Exemplary results for number 2

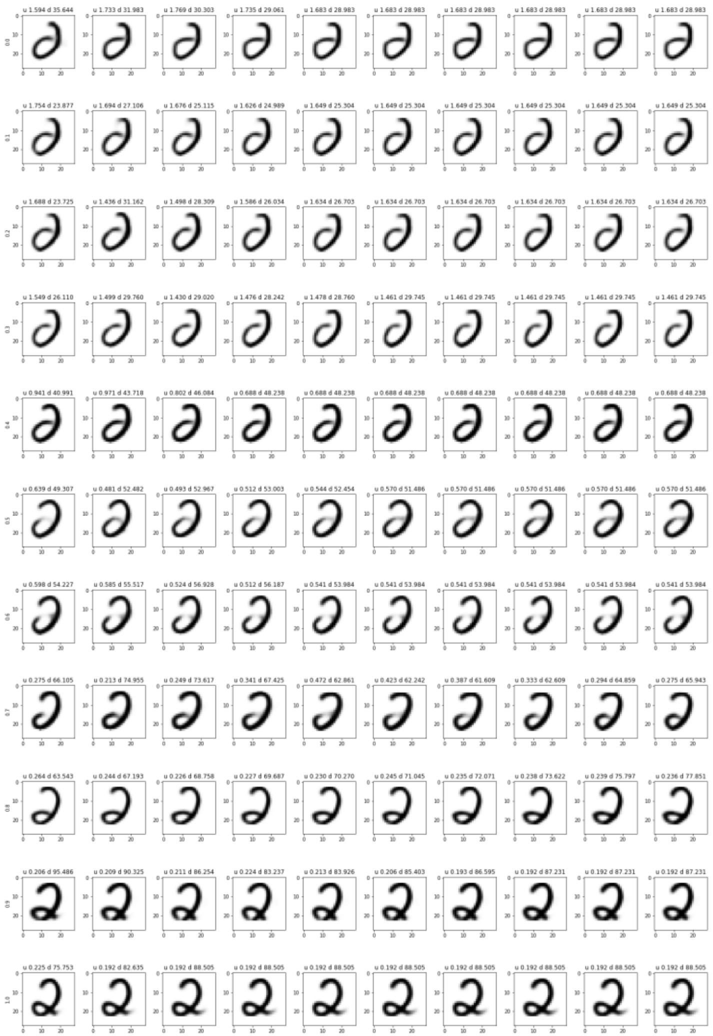


Original sample,
predicted label 3

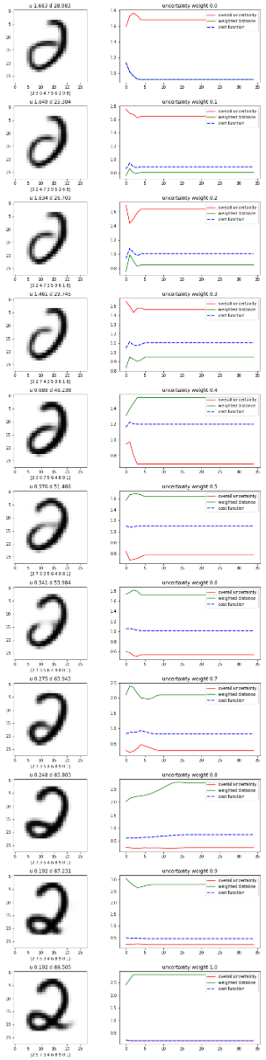


iterations

λ



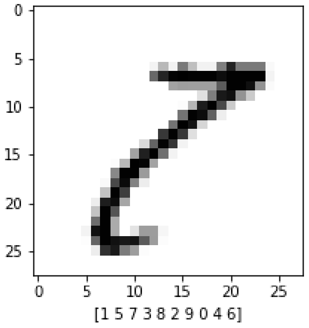
…



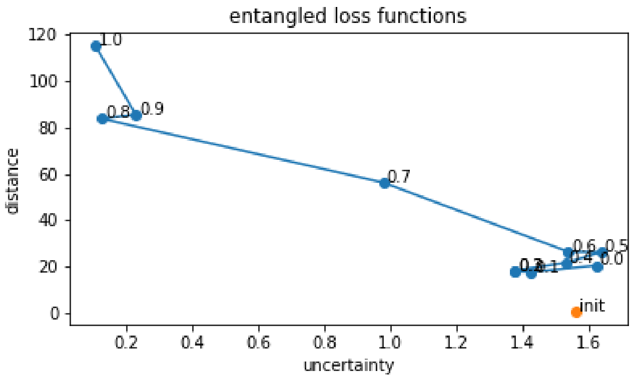
label:

2
0
0
0
2
2
2
2
2
2
2

Exemplary results for number 7

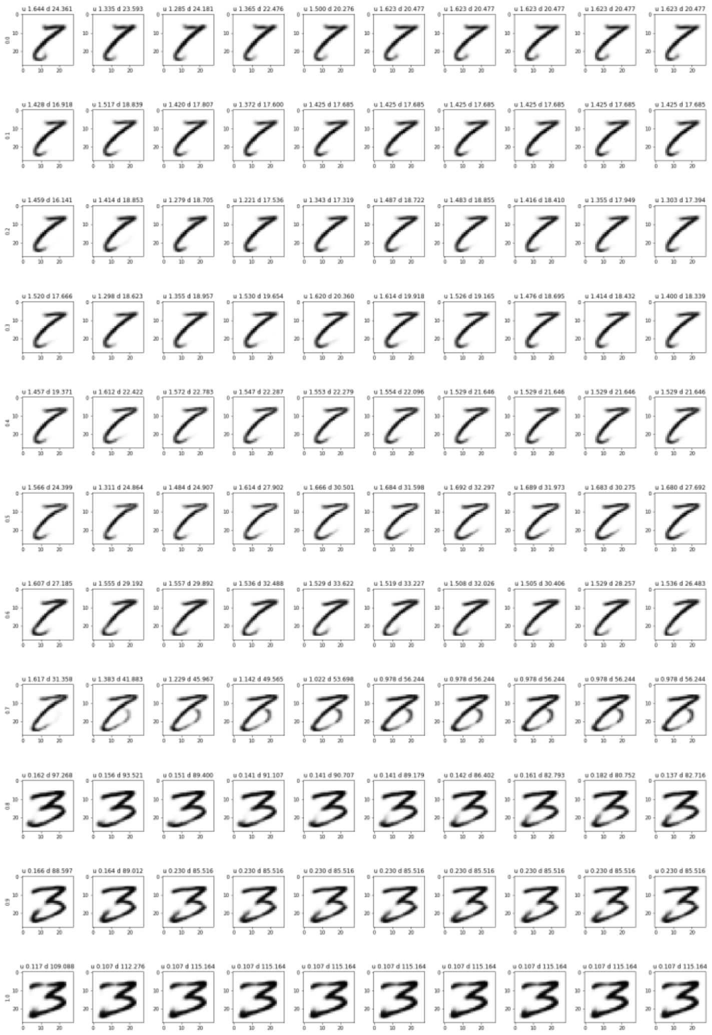


Original sample,
predicted label 1

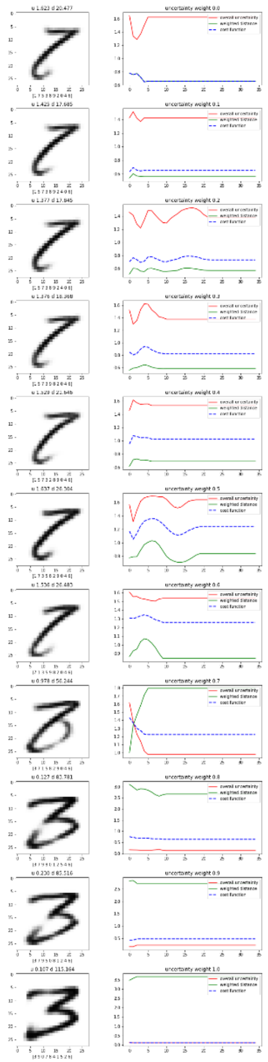


iterations

λ



…

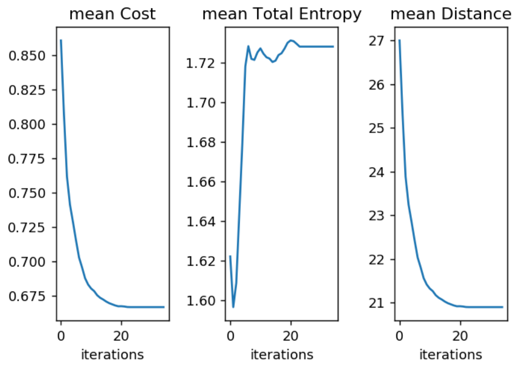


label:

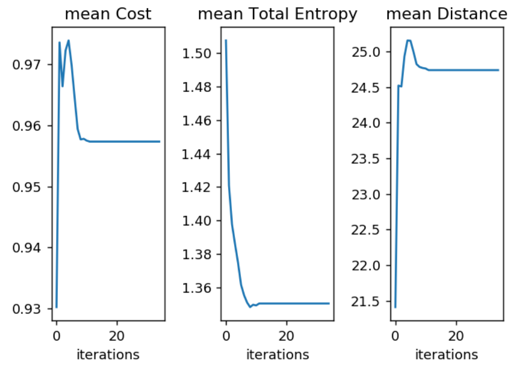
1
1
1
1
1
1
7
3
3
3
3

Loss functions and interpretation

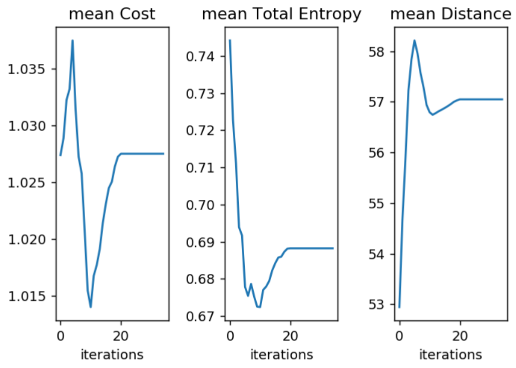
$\boldsymbol{\lambda = 0.0}$



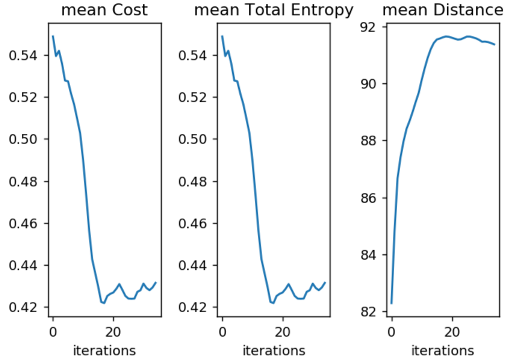
$\boldsymbol{\lambda = 0.3}$



$\boldsymbol{\lambda = 0.7}$



$\boldsymbol{\lambda = 1.0}$



- More weight on distance: more similar CLUEs, often no change in predicted label
- More weight on uncertainty: less similar CLUEs, change in label possible

- local solutions!
- first iteration (1 step of Adam) step brings most visually
- too much weight to one objective often not that useful

Outlook

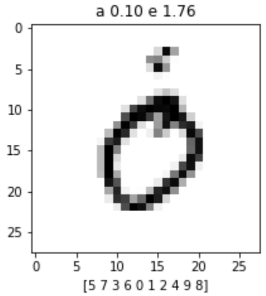
- More extensive studies, different data sets

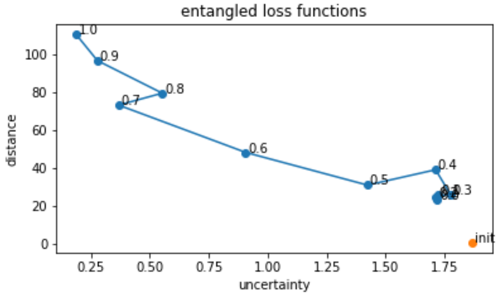
- Solving methods for multiobjective optimization problems
  - Local:
    - Better initialization/starting points for CLUE → make use of previous computed solutions
    - Stochastic gradient descent methods for multiobjective problems (?)

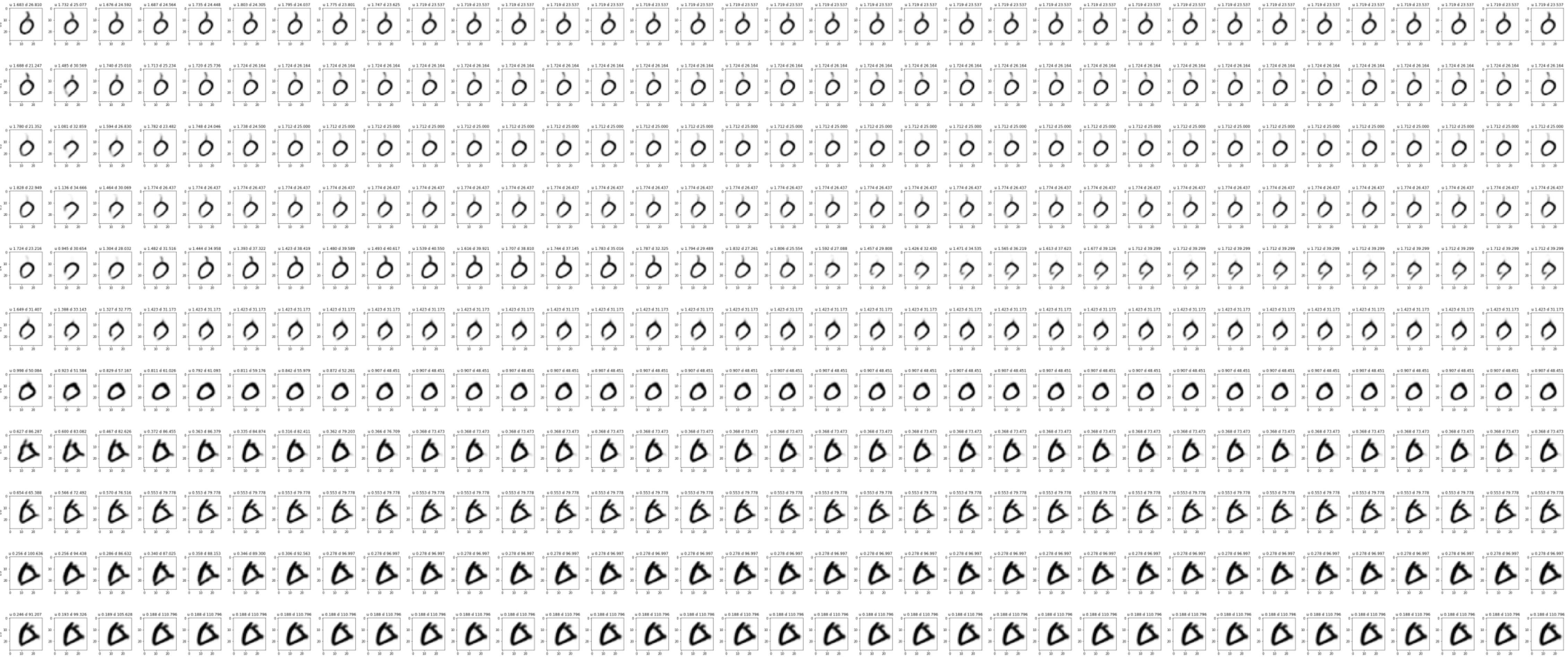
  - Global: Evolutionary algorithms, e.g. NSGA-II

Thank you for your attention!

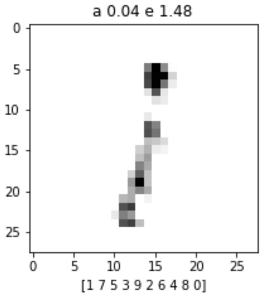
Further examples

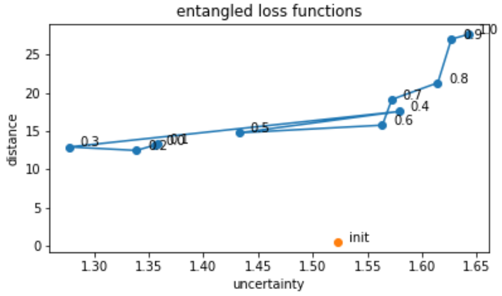


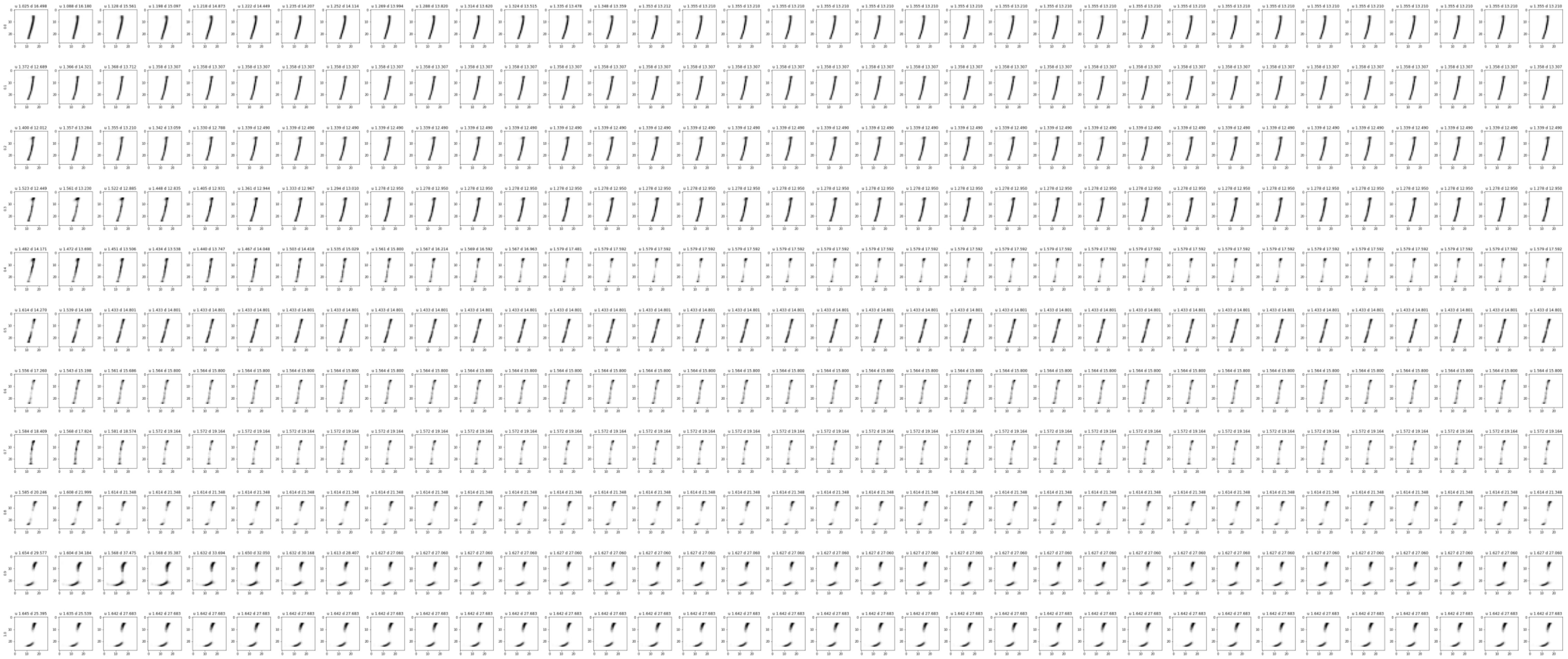




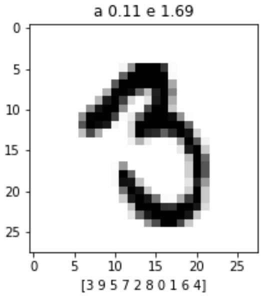
Further examples

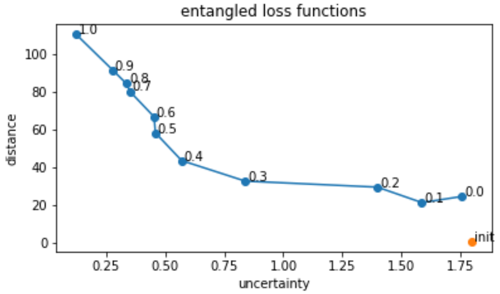


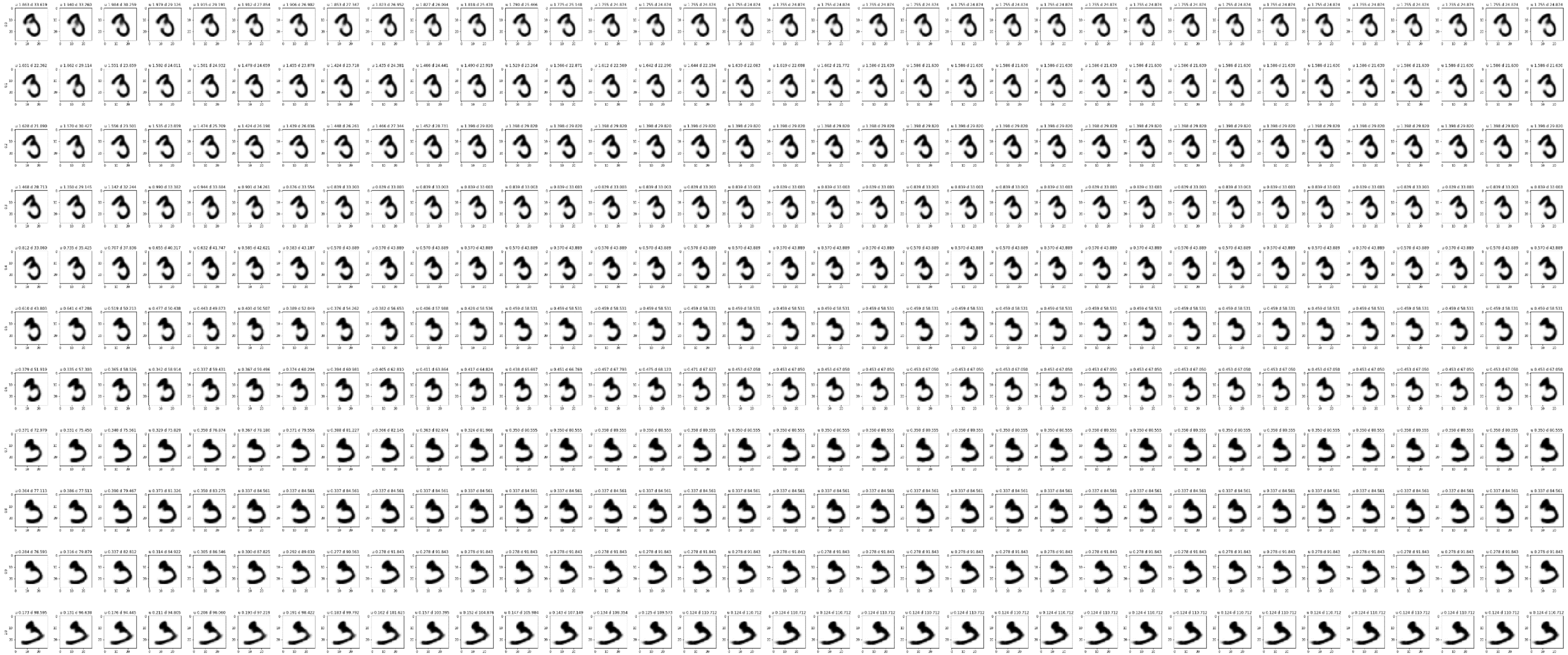




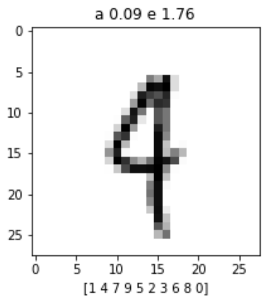
Further examples

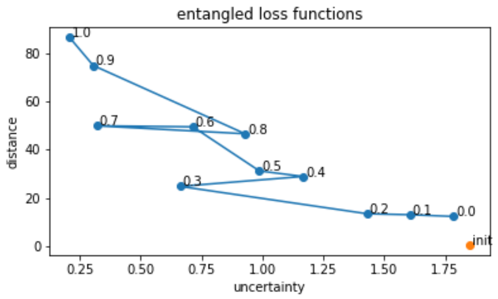


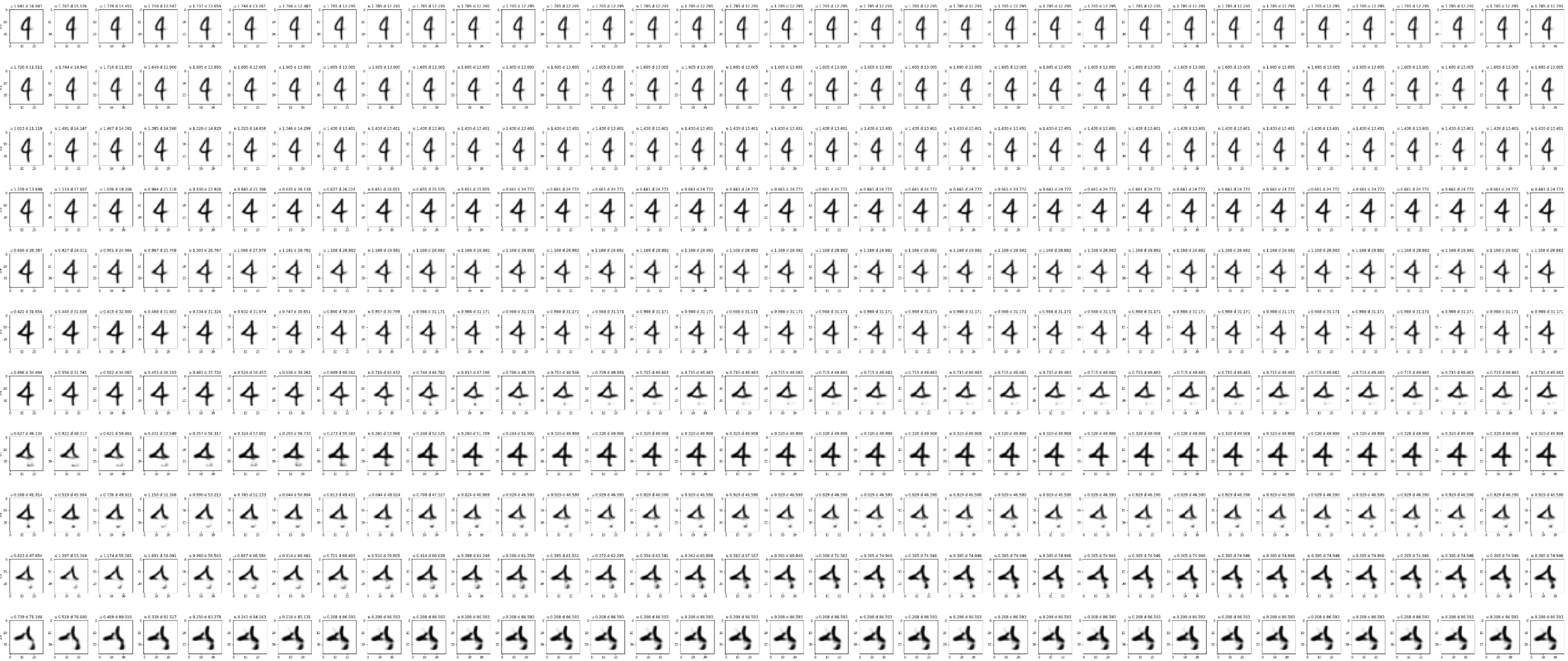




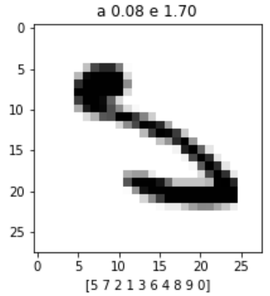
Further examples

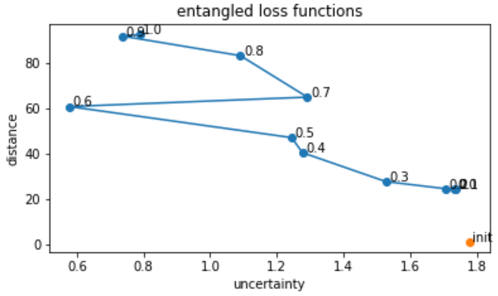


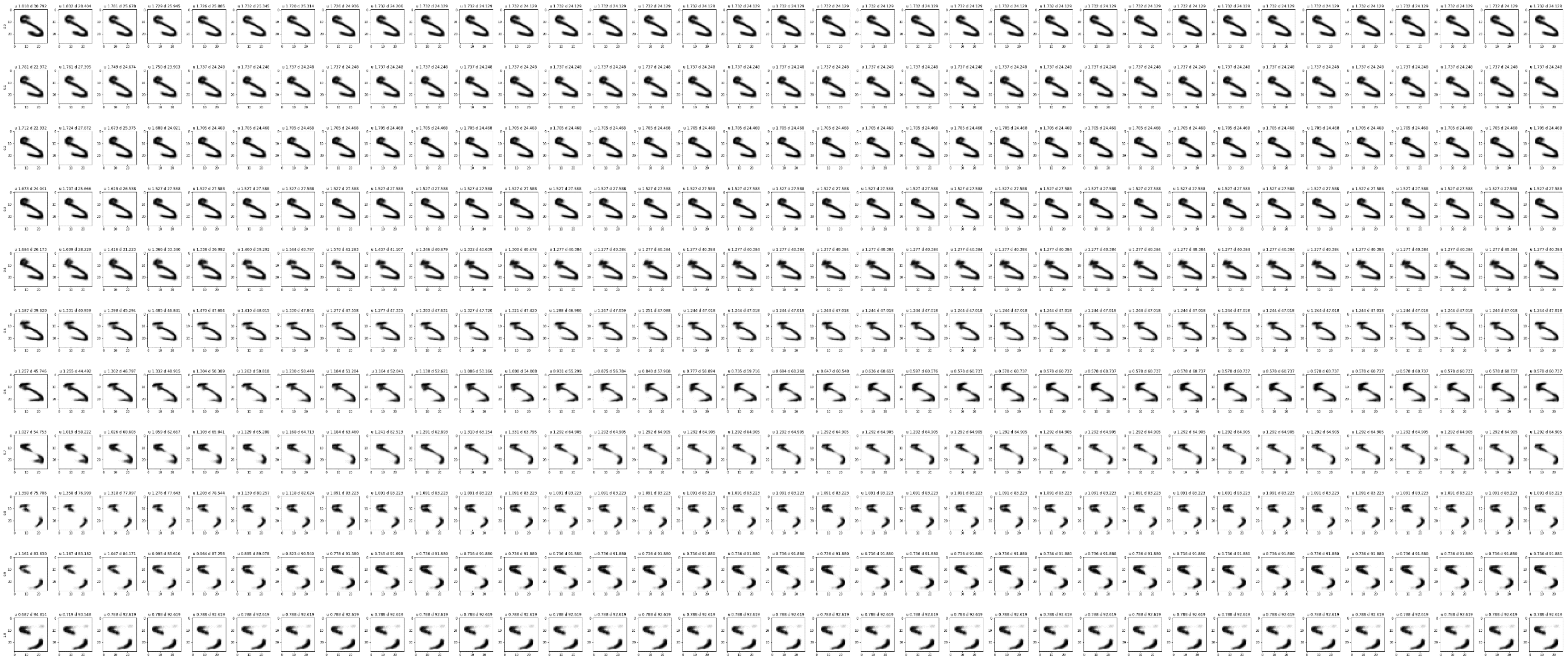




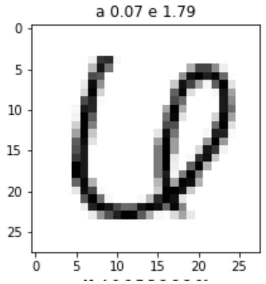
Further examples

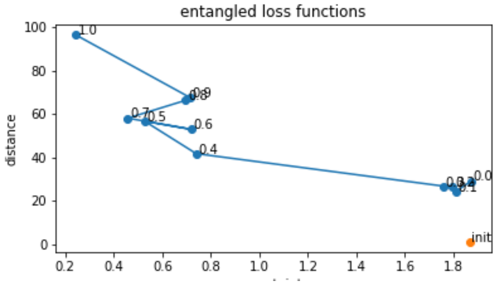


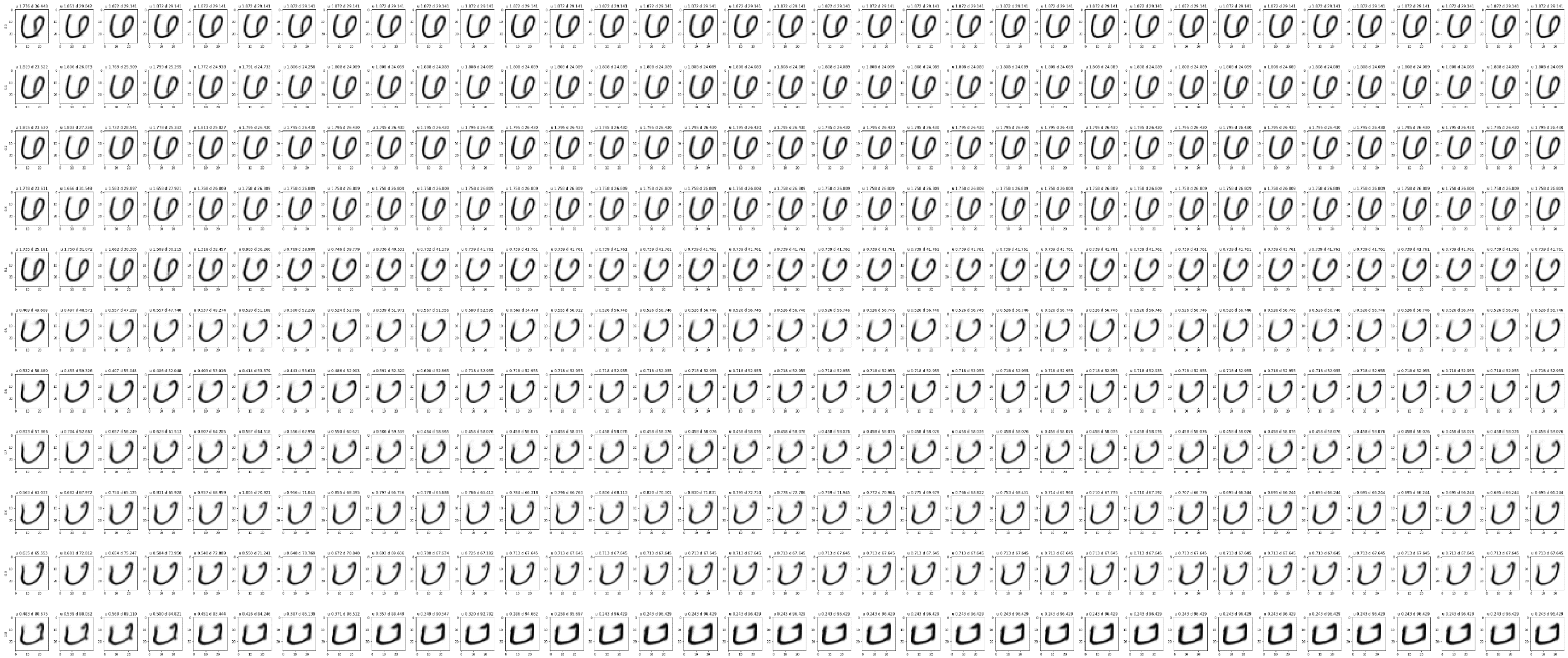




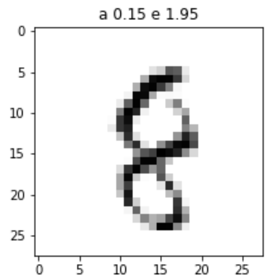
Further examples

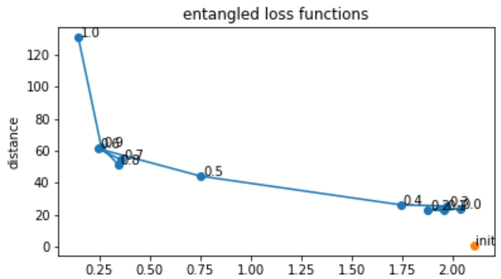


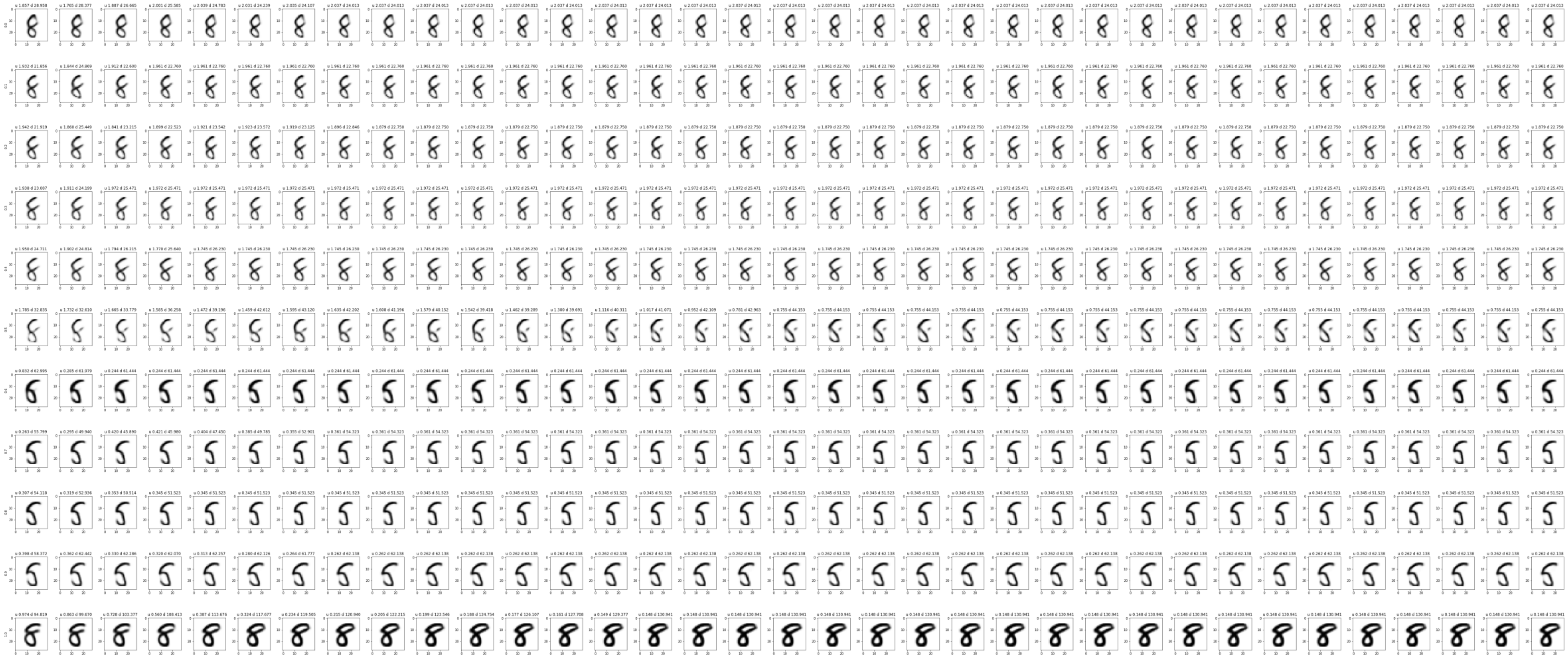




Further examples







Further examples