

Main results optimizing 10d regression functions fitting a student-t distribution with **ACT stick-breaking** model

Performance of "Meta" and "ACT-SB" models on 10,000 freshly sampled regression functions using a student-t distribution to fit the data points and derive the regression parameters.

All models consist of an RNN network (LSTM cells) with 2 layers and a hidden state size of 40 units.

Models were trained for 50 epochs on 10,000 newly sampled functions per epoch (using a batch size of 125). The models were evaluated each 5th epoch on a fixed set of 10,000 validation functions. We picked the best model for the final test run.

Finally the models were evaluated on 10,000 newly sampled functions and unrolled for **500** time steps.

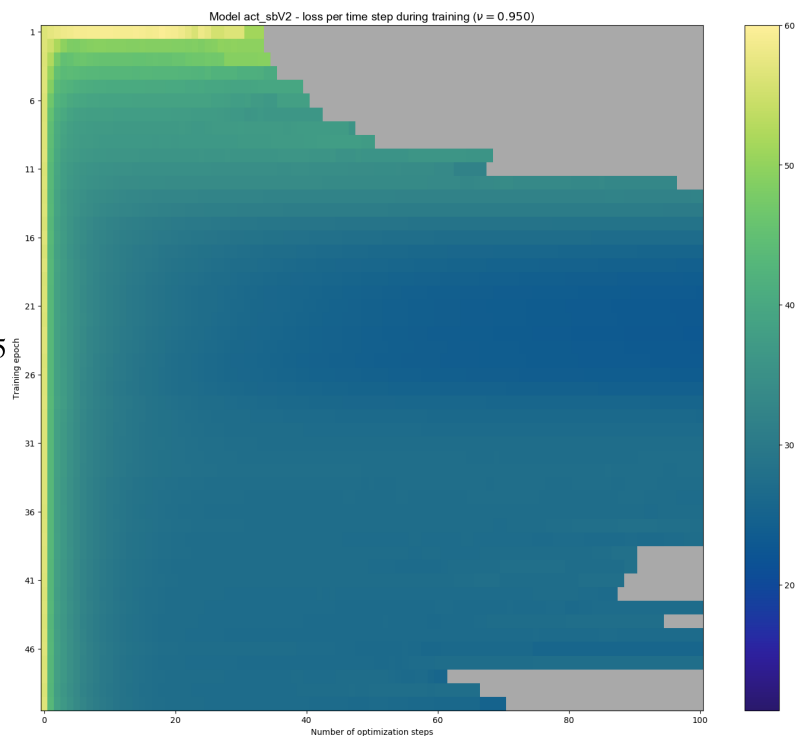
Please note that during training the ACT-SB models were unrolled for a maximum of 100 time steps in case the model did not "cross" the uniformly sampled threshold for a particular optimizee (thresholds were sampled for each optimizee) in order to prevent the model from taking too many steps. We increased the maximum number of time steps to 200, but this did not change the overall results of the models.

More specific the following models were evaluated:

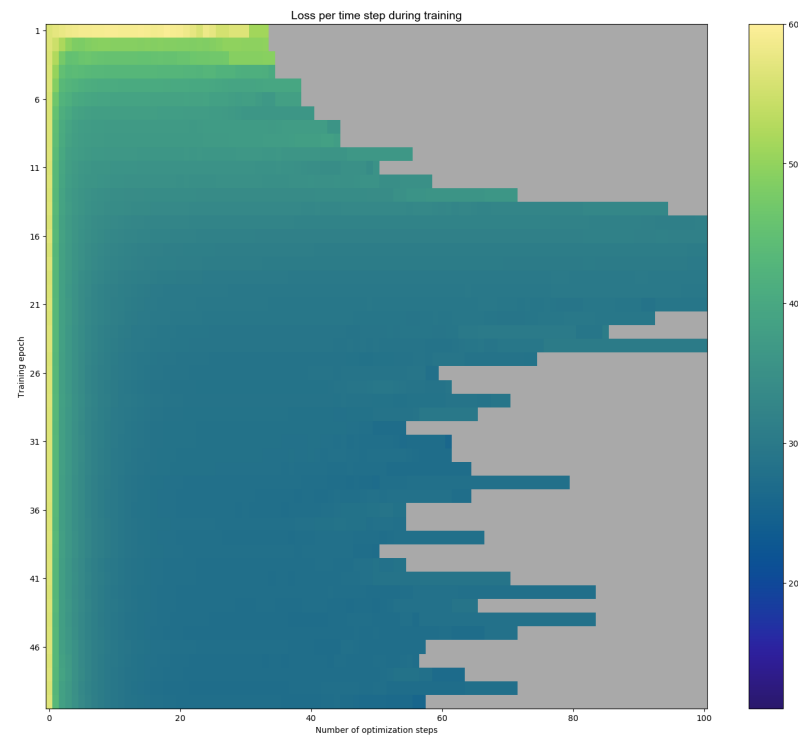
- (1) **metaV1**: baseline model from L2L paper where each mini-batch of optimizees is trained for a horizon of 50 time steps ($\text{lr}=3\text{e-}5$).
- (2) **metaV2**: baseline model that uses a stochastic training regime i.e. the horizon for a mini-batch is sampled from a distribution $p(T)$ with $E[T]=26$ optimization steps ($\text{lr}=3\text{e-}5$).
- (3) **act_sbV1**: the **extended** baseline model that in addition to the delta parameter values of the optimizees (functions to be optimized) generates the loss-weights which are interpreted in the model-context as probabilities and referred to as *qt-values* (denoted $q(t | x)$), the probability of performing t time steps i.e. to stop after t steps. In order to generate the *qt-values* we apply a so called *stick-breaking* procedure ($\text{lr}=5\text{e-}5$).
- (4) **act_sbV2**: stick-breaking ACT that uses a **KL-divergence cost annealing** scheme (logistic shape). All models were trained for 50 epochs and the *kl-weights* were increased during a period of 40 epochs (with final value 1) ($\text{lr}=5\text{e-}5$).

Illustrate *behavior* of ACT-SB models during training with different prior shape parameters (ν) - step loss figures

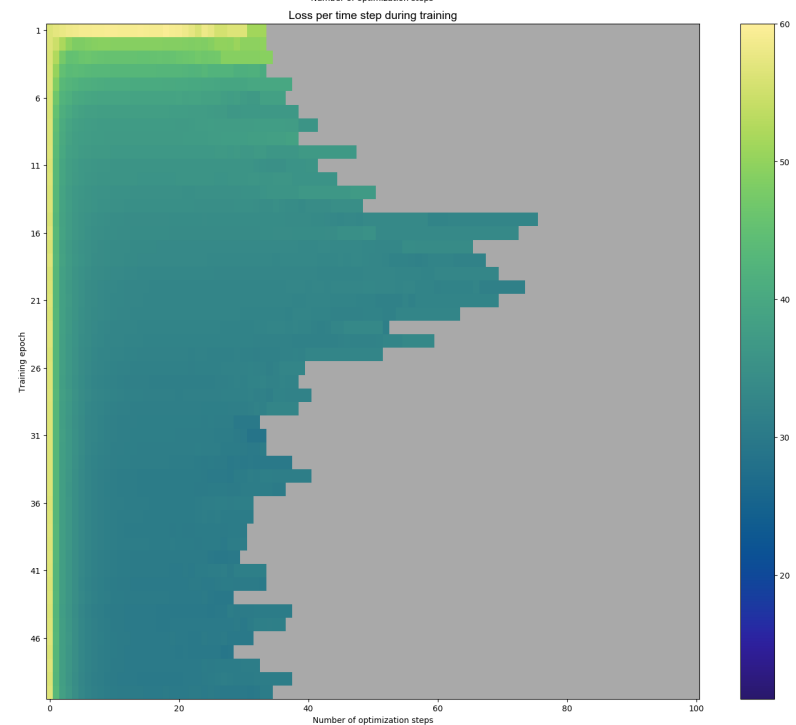
$\nu = 0.95$



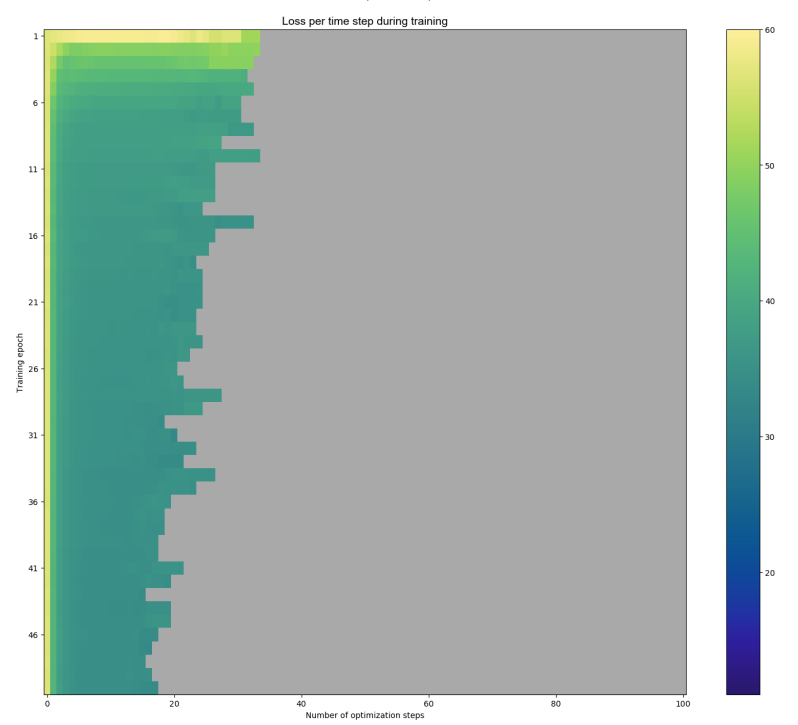
$\nu = 0.9$



$\nu = 0.7$



$\nu = 0.3$

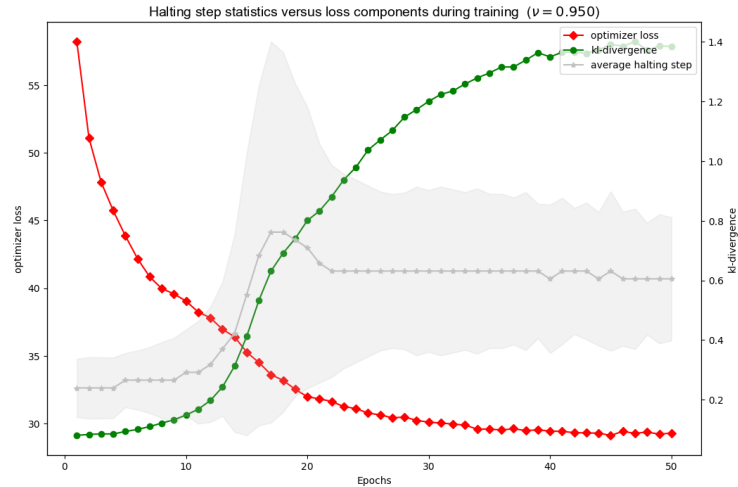


Illustrate *behavior* of ACT-SB models during training with different prior shape parameters (ν)

optimizer loss - KL term - average halting step

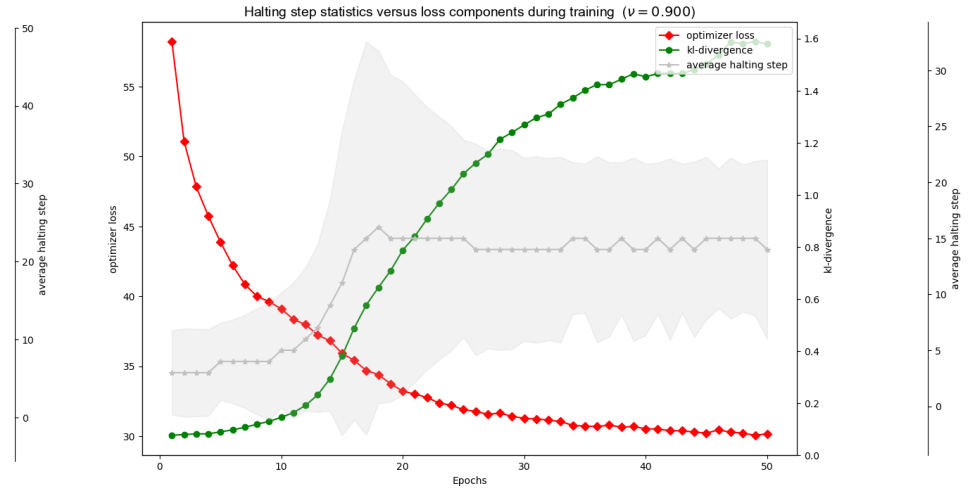
Note: each figure contains 3 y-axis (1) optimizer loss (2) KL divergence (3) average number of halting step (grey area is +/- one stddev). Values are shown per trainings epoch. Each figure captures a different act_sb model trained with a specific prior shape parameter value ν .

$\nu = 0.95$



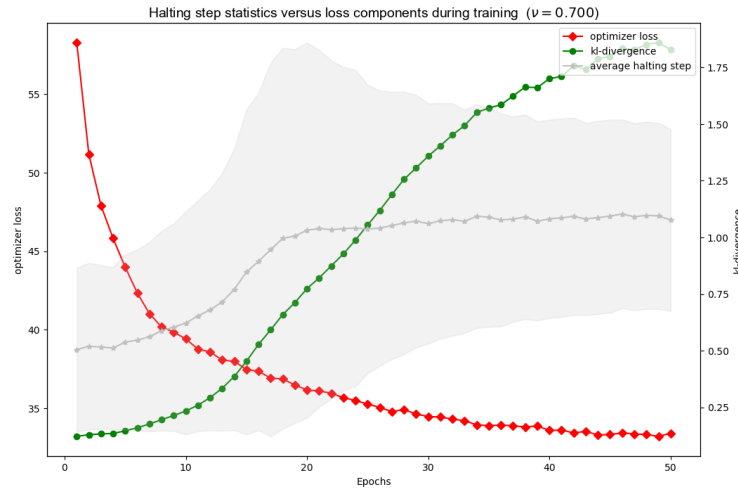
Halting step statistics final epoch: Range(1, 72) / mean=17.0 / stddev=7.9 / median=16.0

$\nu = 0.9$



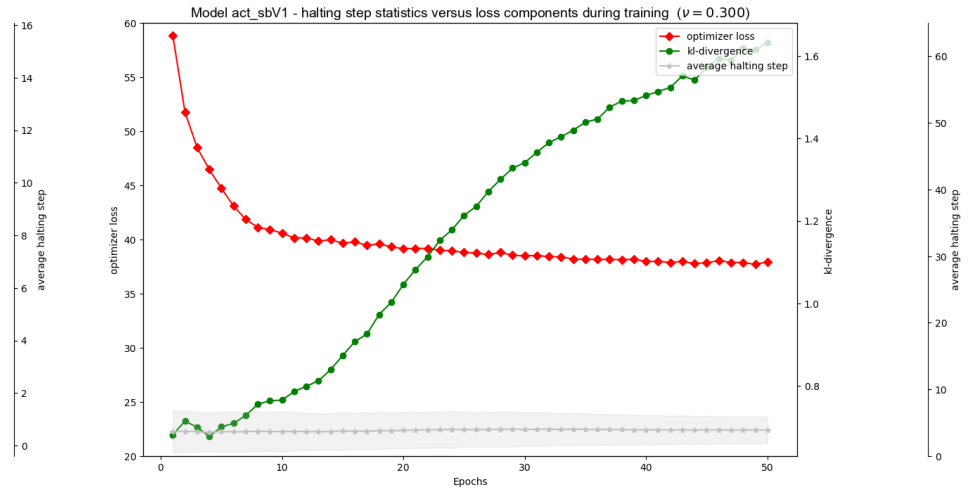
Halting step statistics final epoch: Range(1, 57) / mean=14.0 / stddev=8.0 / median=14.0

$\nu = 0.7$



Halting step statistics final epoch: Range(1, 34) / mean=8.6 / stddev=3.5 / median=8.0

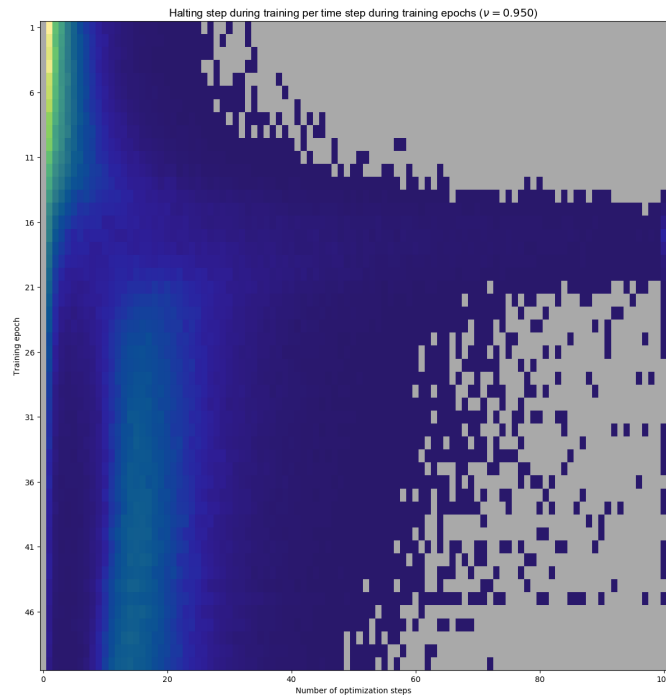
$\nu = 0.3$



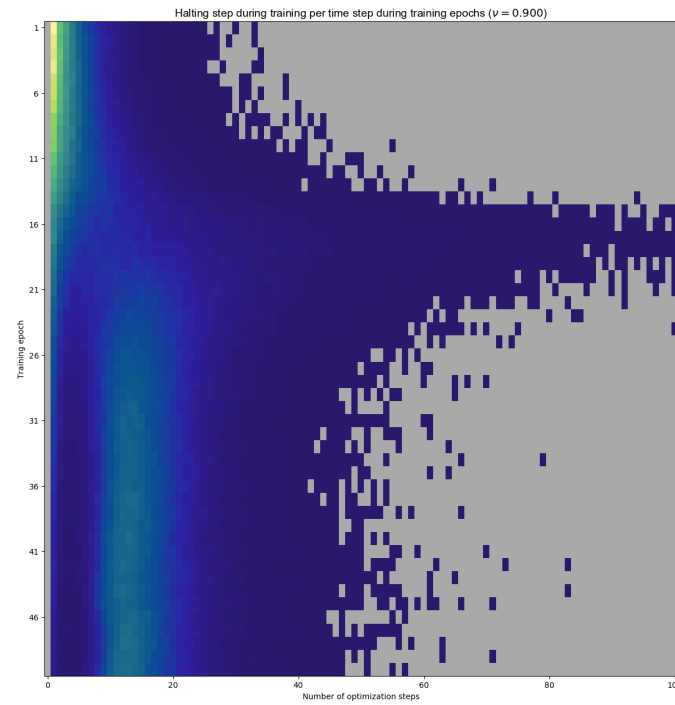
Halting step statistics final epoch: Range(1, 17) / mean=3.9 / stddev=2.0 / median=4.0

Illustrate *behavior* of ACT-SB models during training with different prior shape parameters (ν) halting step distributions

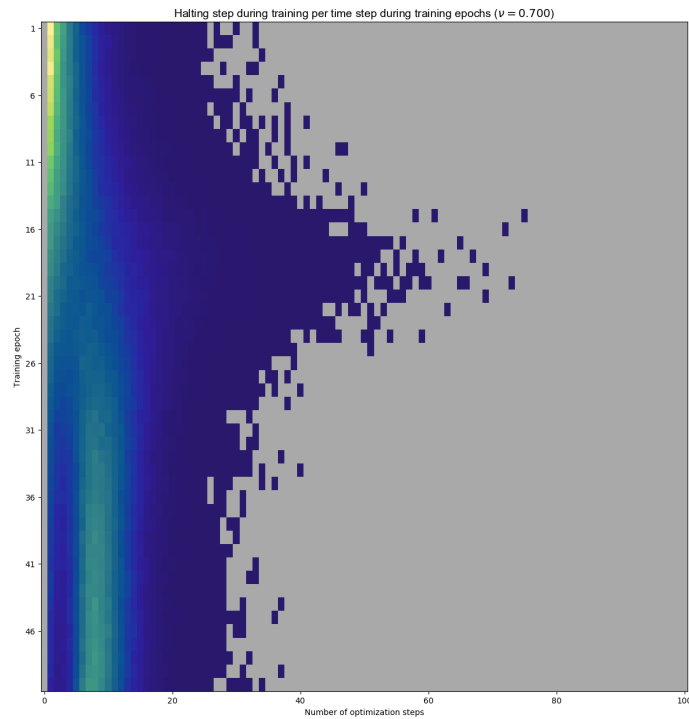
$\nu = 0.95$



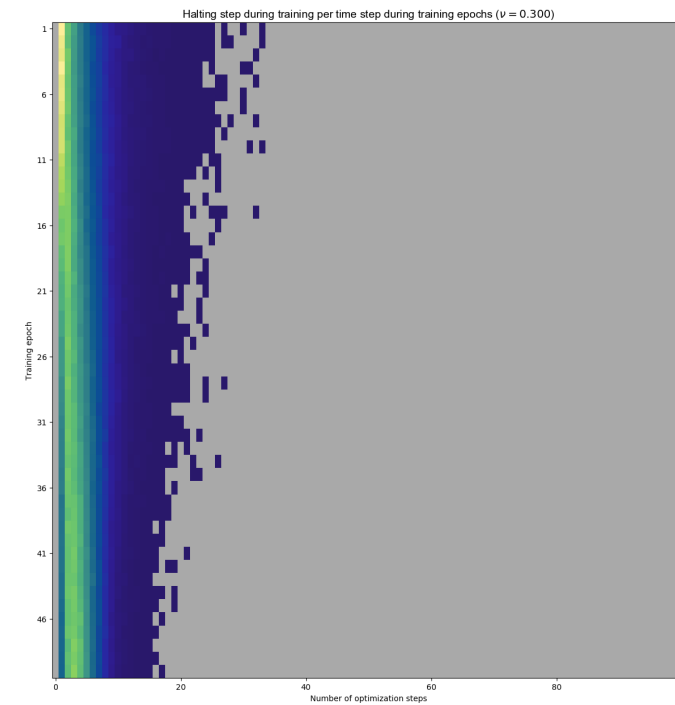
$\nu = 0.9$



$\nu = 0.7$



$\nu = 0.3$



Compare ACT-SB with shape parameter **0.95** with Meta models **V1**($T=50$) and **V2** ($E[T]=26$)

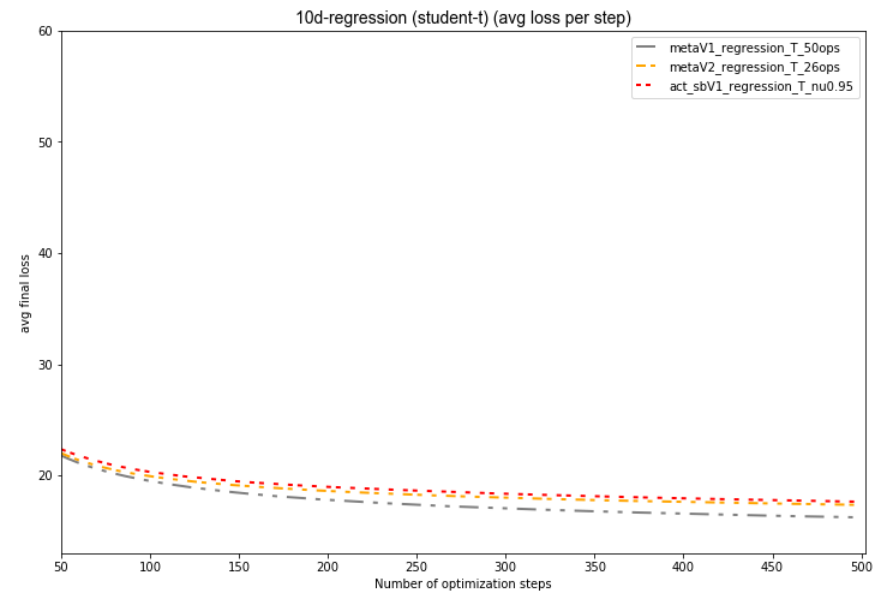
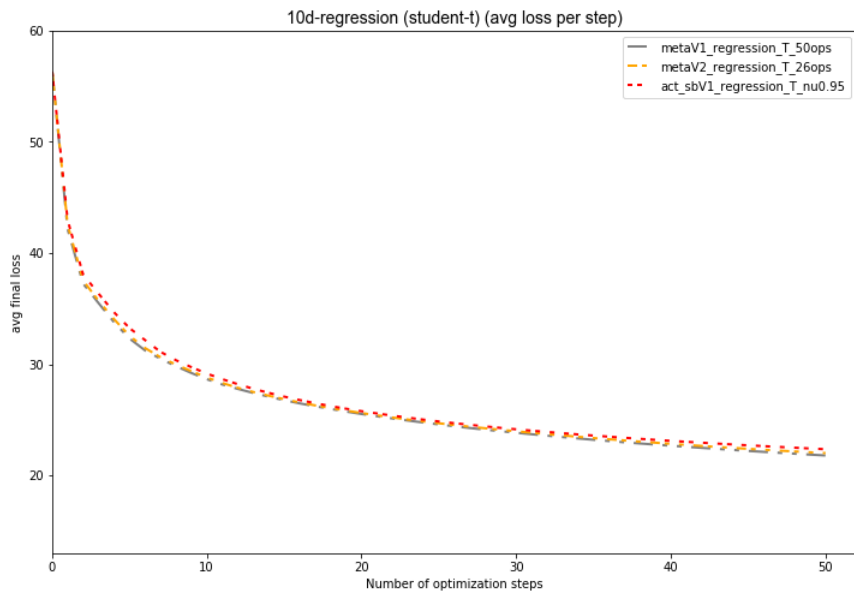
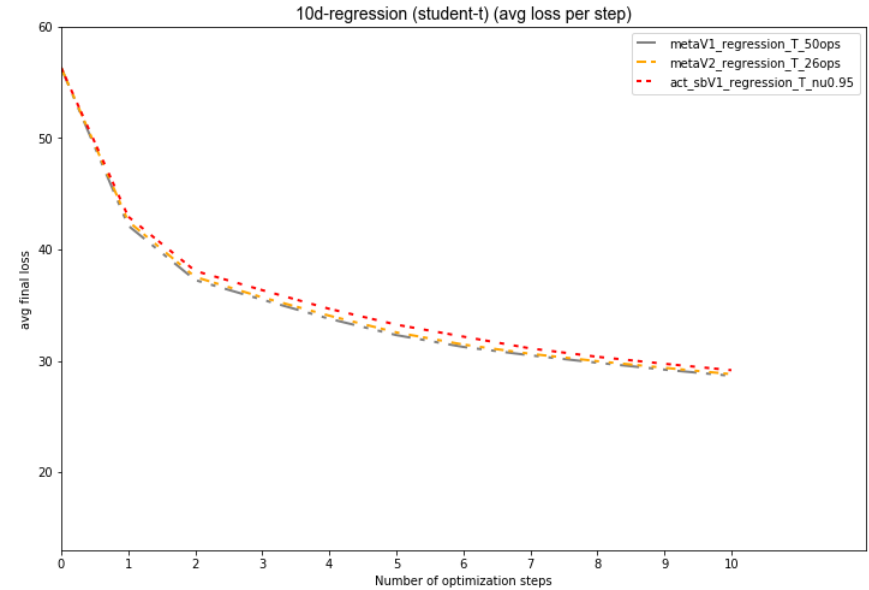
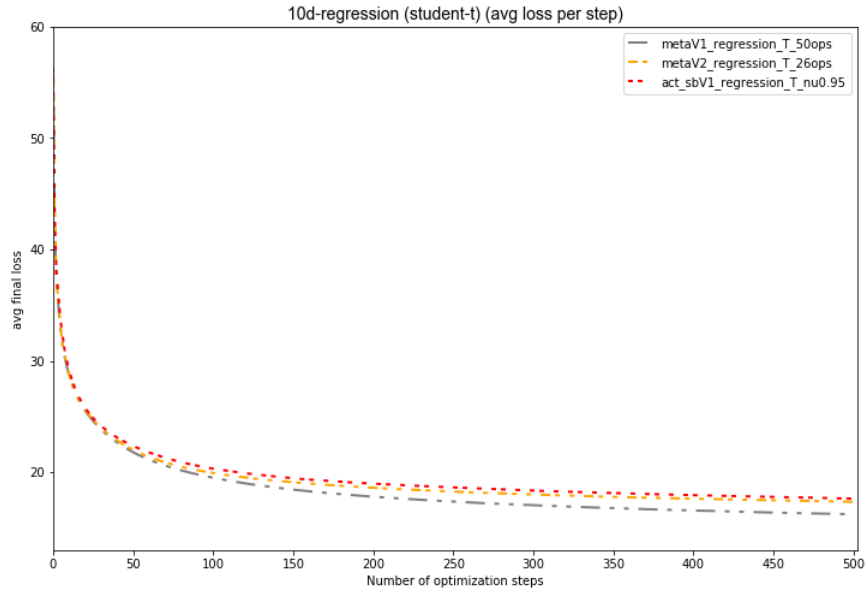
Note:

(1) figures show performance of the same models on 10,000 test functions. Each figure captures different number of time steps (0-500, 0-10, 0-50 and 50-500)

Take-aways: (a) act-sb model with shape parameter 0.95 performs roughly the same as meta model trained with a stochastic training regime with $E[T]=26$;

(b) after 50 time steps performance of act_sb models gets inferior compared to **meta model V1** trained for 50 time steps (fixed horizon);

(c) compared to act-sb models with smaller shape parameter, this models achieve the best optimization performance.



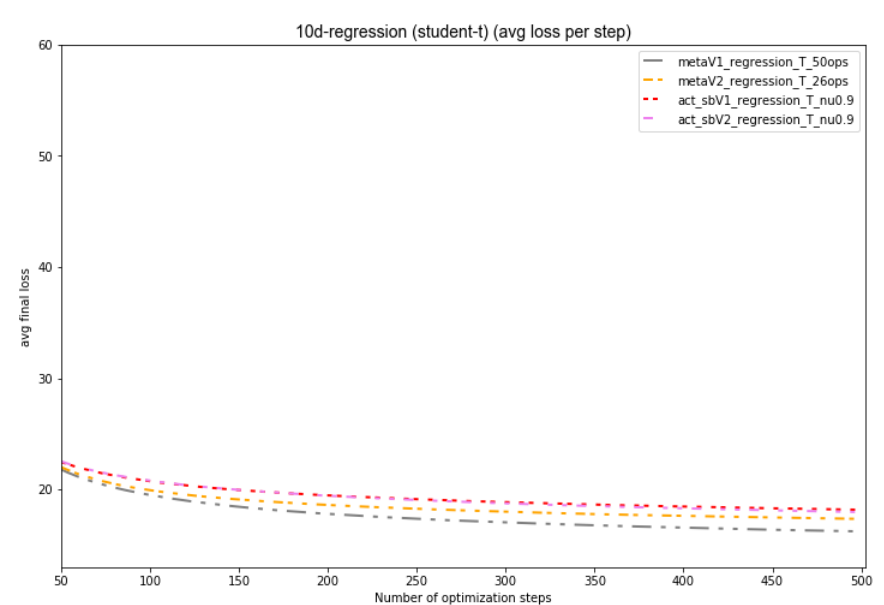
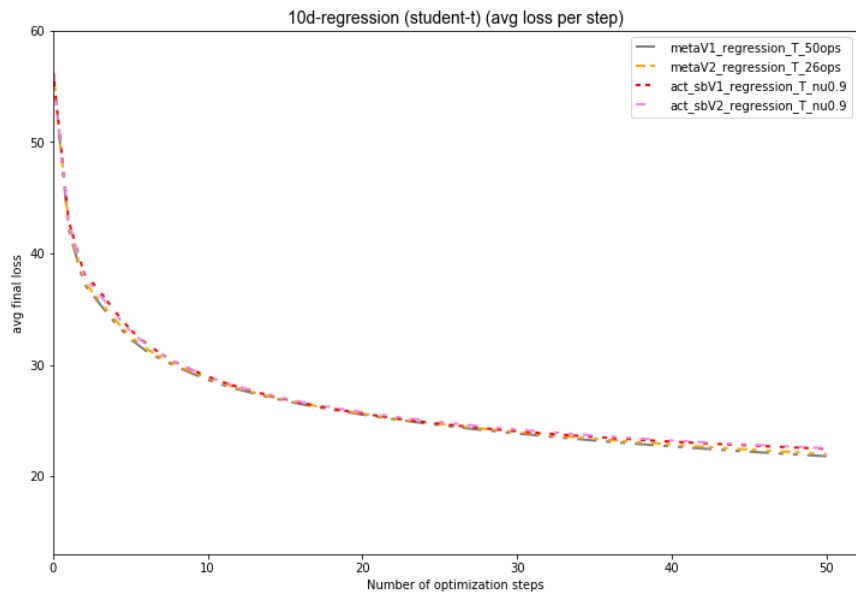
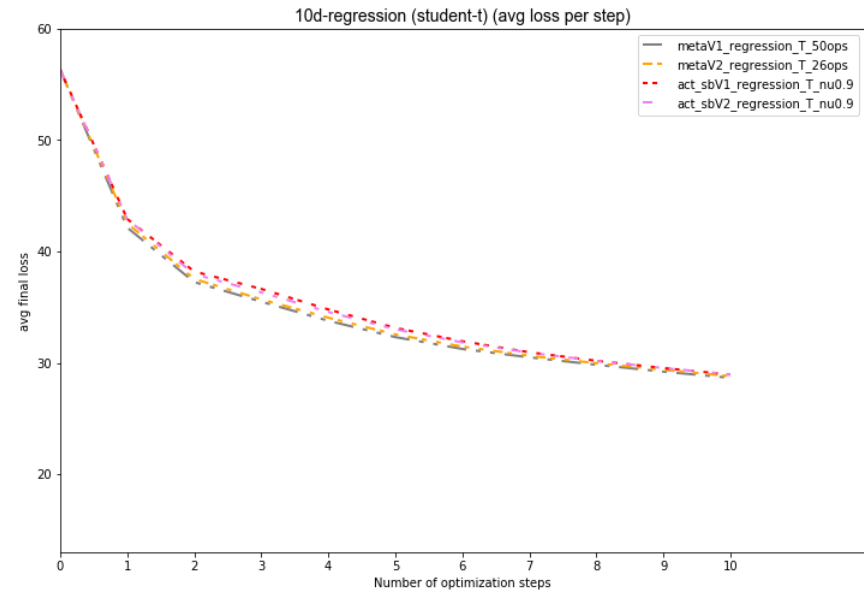
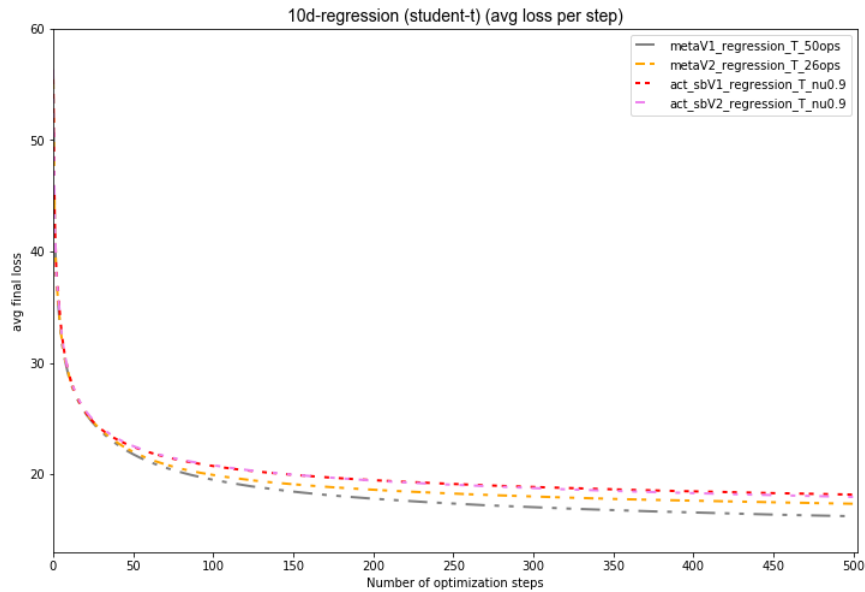
Compare ACT-SB with shape parameter **0.90** with Meta models **V1**(T=50) and **V2** (E[T]=26)

Note:

(1) figures show performance of the same models on 10,000 test functions. Each figure captures different number of time steps (0-500, 0-10, 0-50 and 50-500)

(2) act_sbV2 model used a **KL divergence cost annealing** schedule (logistic form) during the first 40 epochs during training

Take-aways: (a) KL annealing schedule does not influence performance; (b) after 50 time steps performance of act_sb models gets inferior compared to meta models.

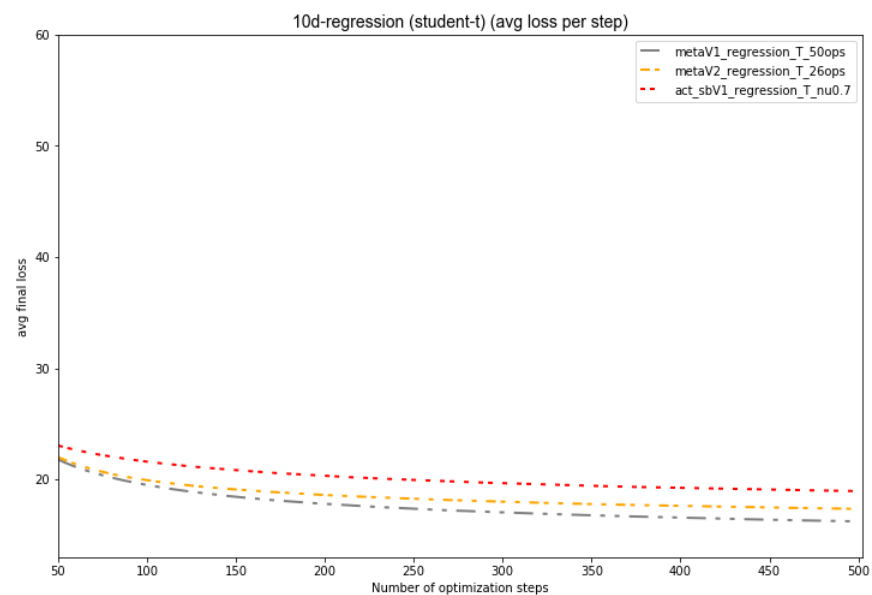
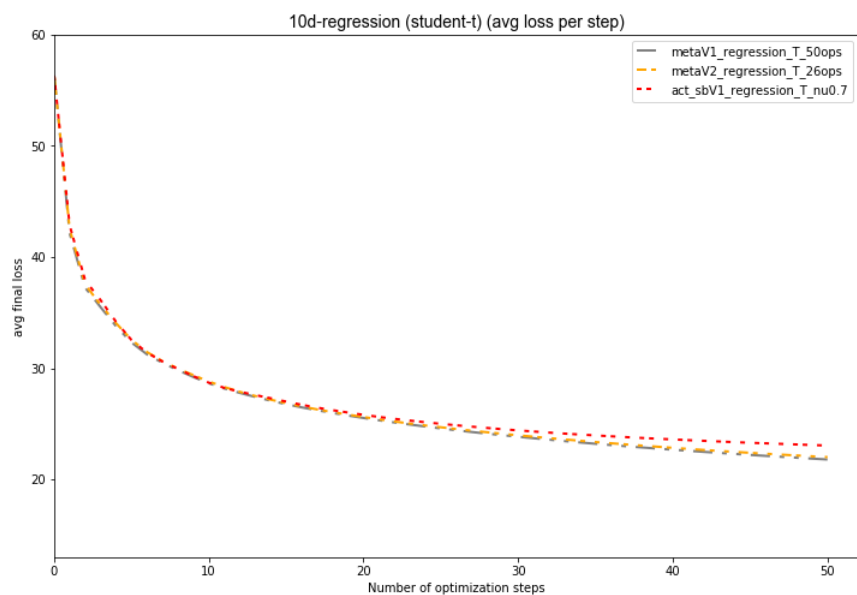
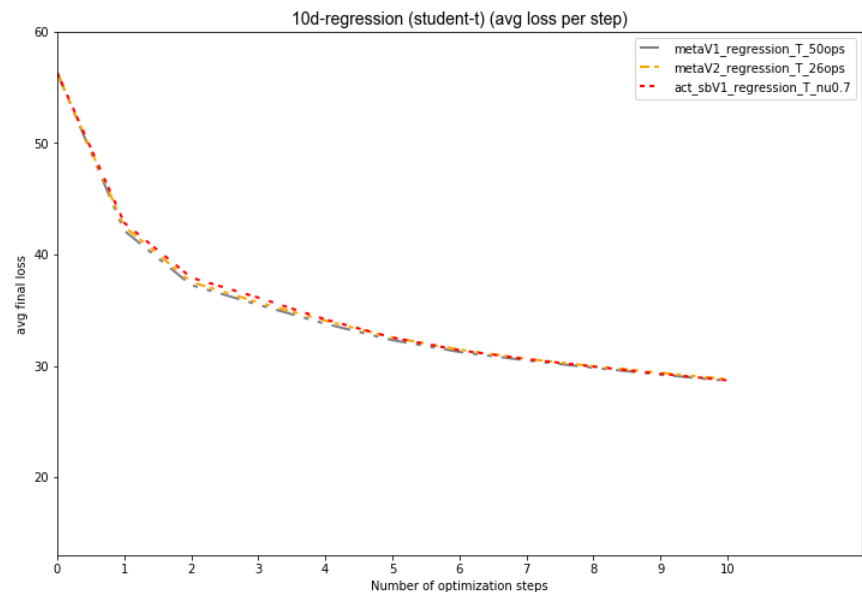
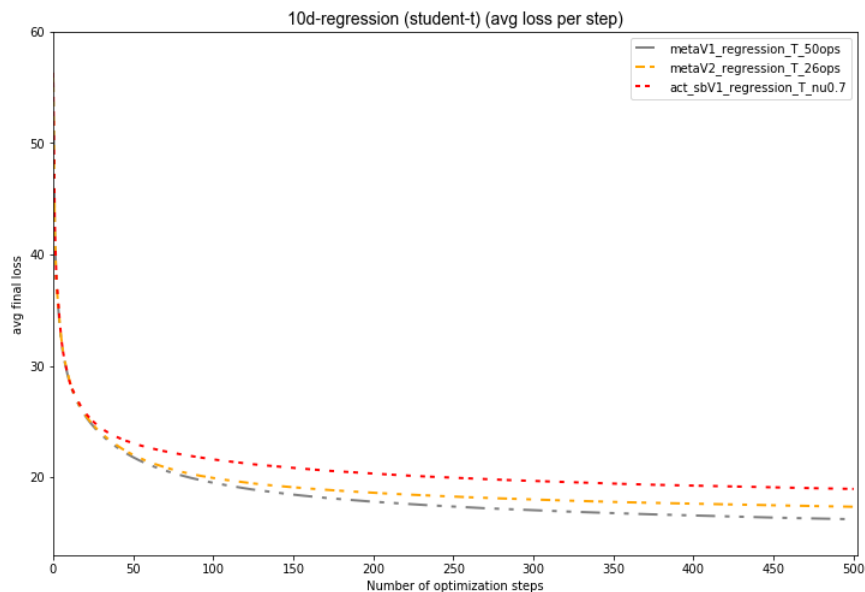


Compare ACT-SB with shape parameter **0.7** with Meta models **V1**($T=50$) and **V2** ($E[T]=26$)

Note:

(1) figures show performance of the same models on 10,000 test functions. Each figure captures different number of time steps (0-500, 0-10, 0-50 and 50-500)

Take-aways: (a) after roughly 30 time steps the performance of the act_sb model deteriorates compared to the meta models.



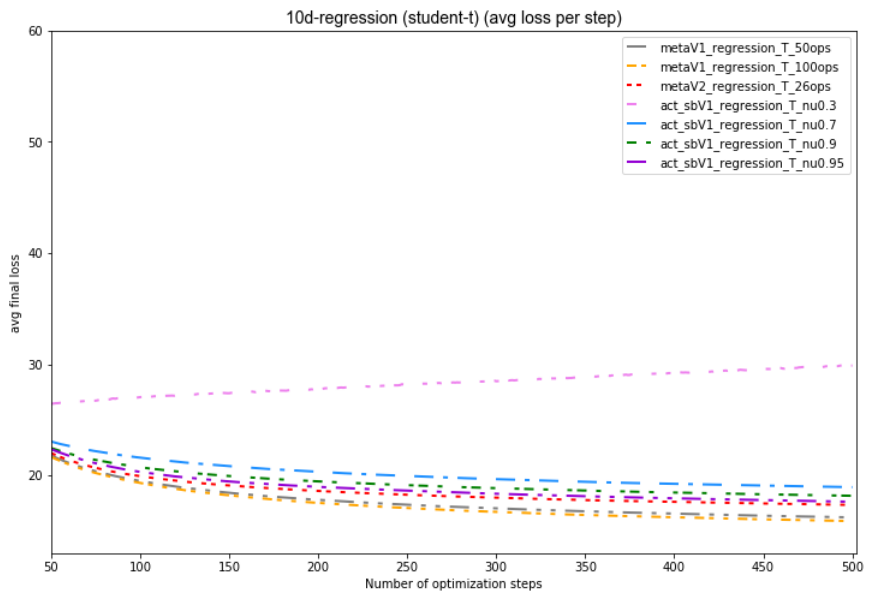
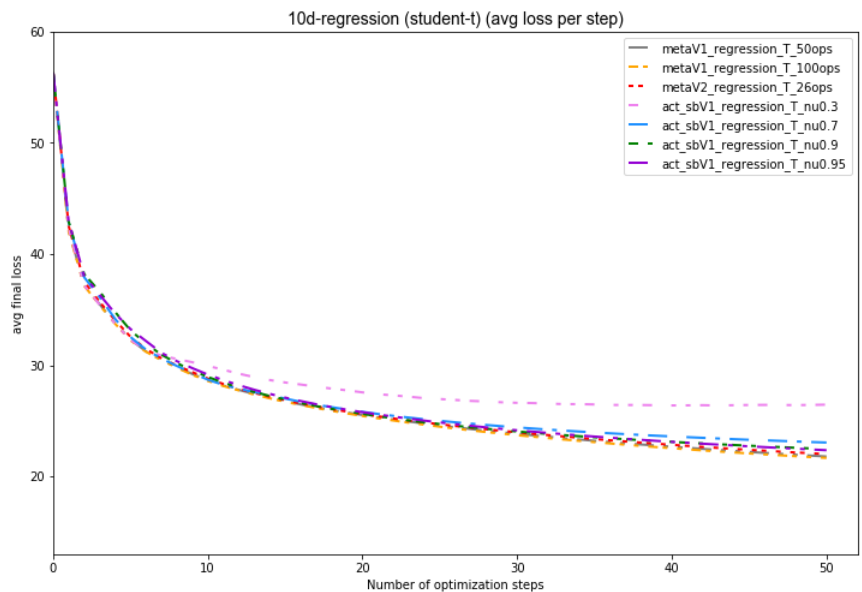
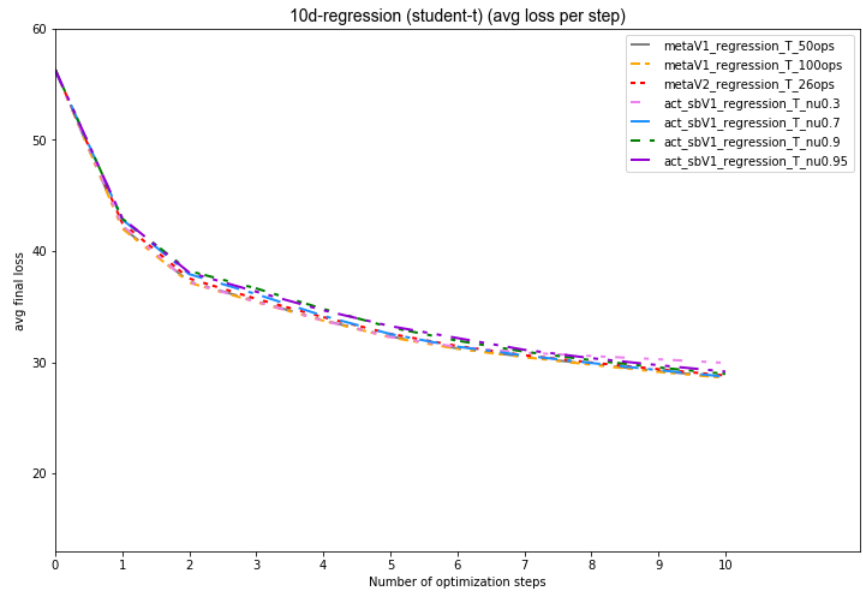
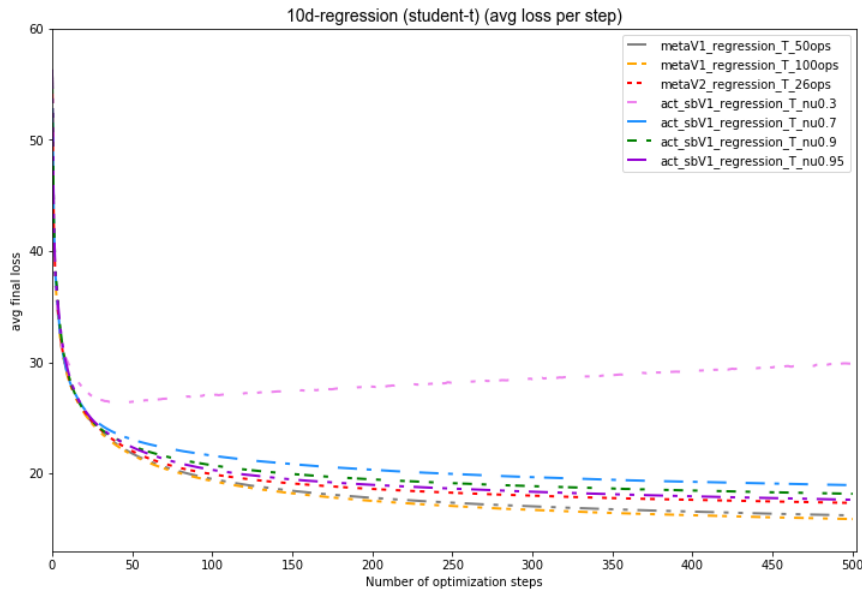
Compare ACT-SB with different shape parameters with Meta models **V1**($T=50$, $T=100$) and **V2** ($E[T]=26$)

Overview of all models tested - step losses over time

Note:

(1) figures show performance of the same models on 10,000 test functions. Each figure captures different number of time steps (0-500, 0-10, 0-50 and 50-500)

Take-aways: (a) act_sb with shape parameter equal to 0.3 deteriorates already after 9 time steps (b) metaV1 model trained with a fixed horizon of 100 time steps performs slightly better than same model trained for 50 time steps; (c) generally, smaller shape parameter size of prior $p(t)$ results in inferior model performance in later time steps;



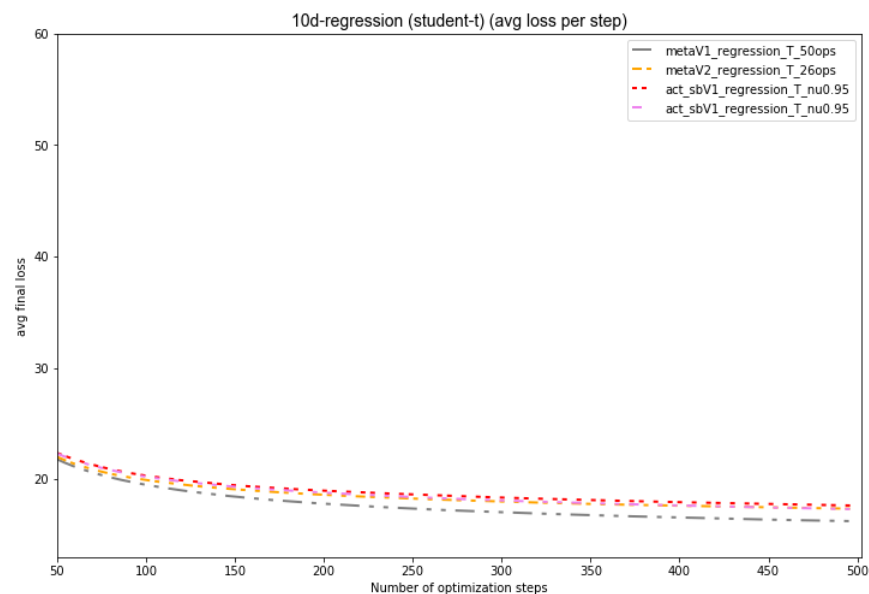
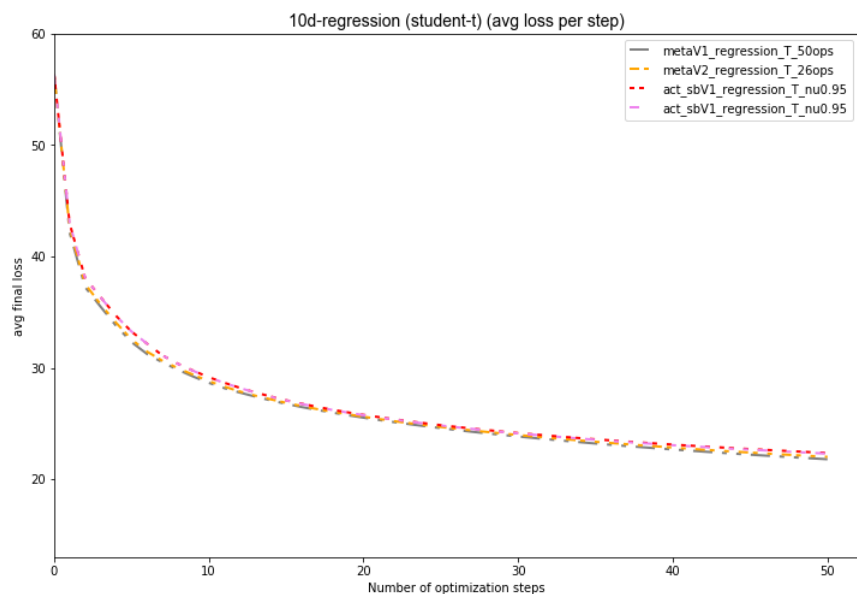
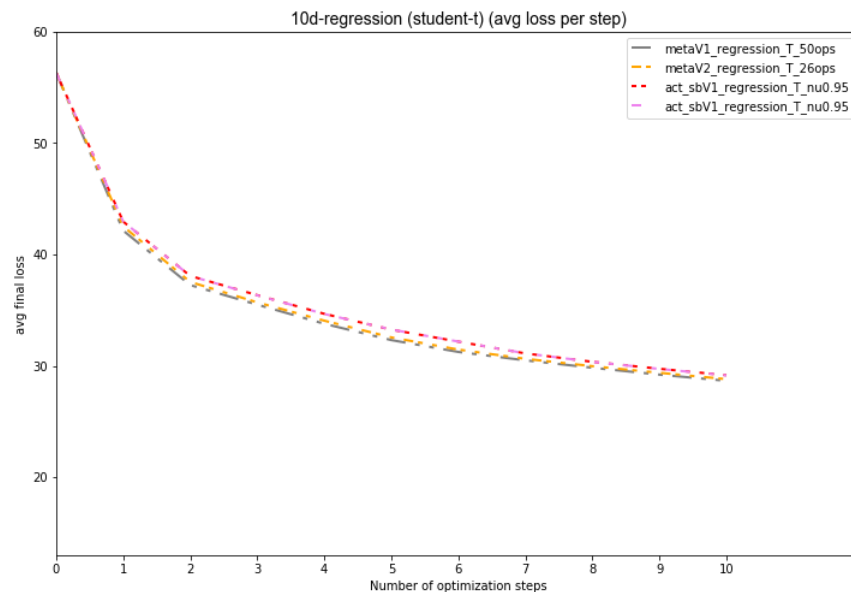
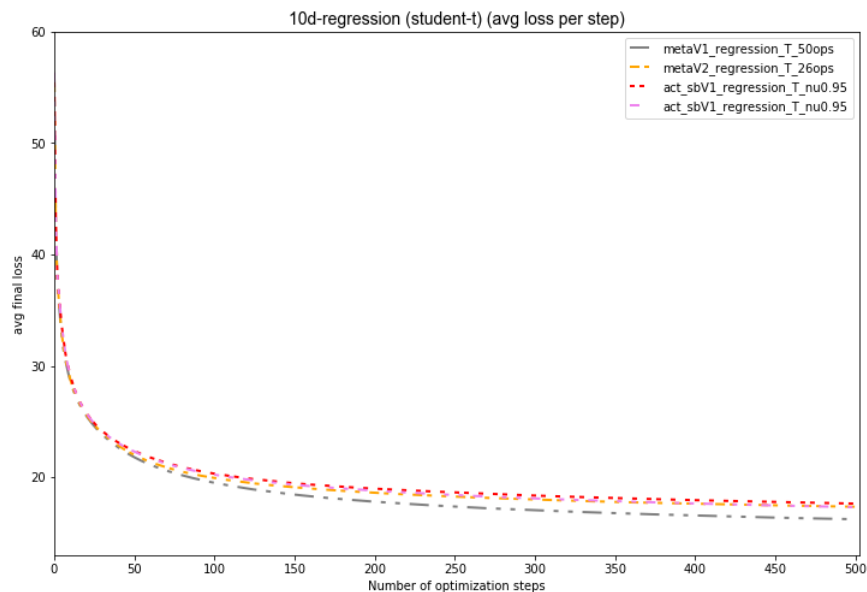
Compare ACT-SB with shape parameter **0.95** with Meta models **V1**(T=50) and **V2** (E[T]=26)

Compare training with max horizon **100** versus **200** time steps

Note:

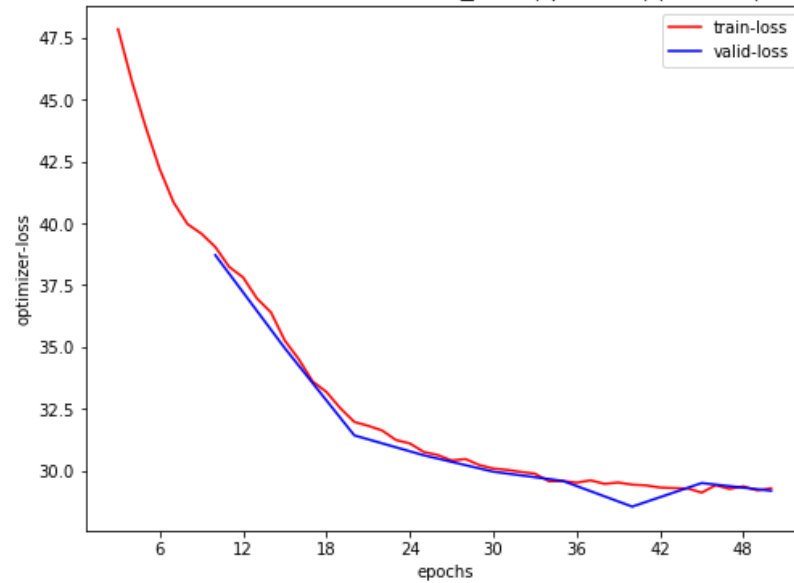
- (1) act_sbV1 with **RED** line was trained with max time step horizon equal to 100 steps, whereas act_sbV1 with **PINK** line was trained with "max T" equal to 200.
- (2) figures show performance of the same models on 10,000 test functions. Each figure captures different number of time steps (0-500, 0-10, 0-50 and 50-500)

Take-away: increasing the maximum number of time steps the model can take during training (from **100** to **200**) results in roughly the same model performance

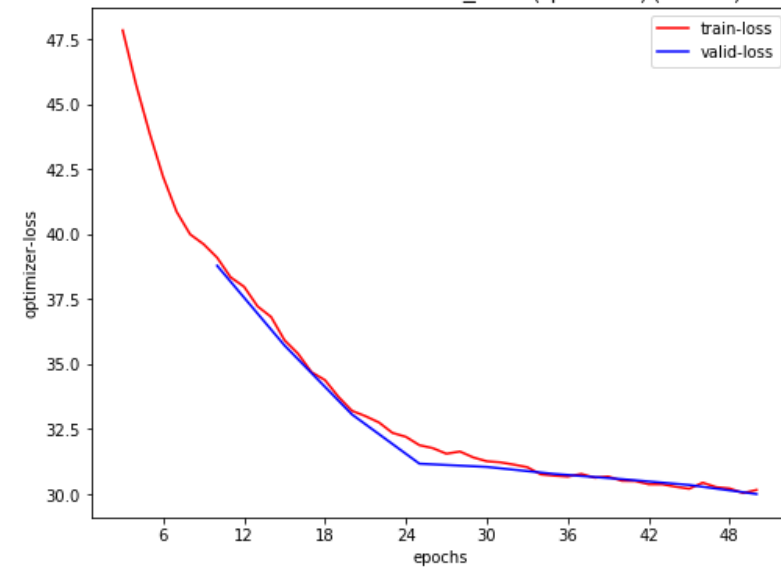


Learning curves ACT-SB models

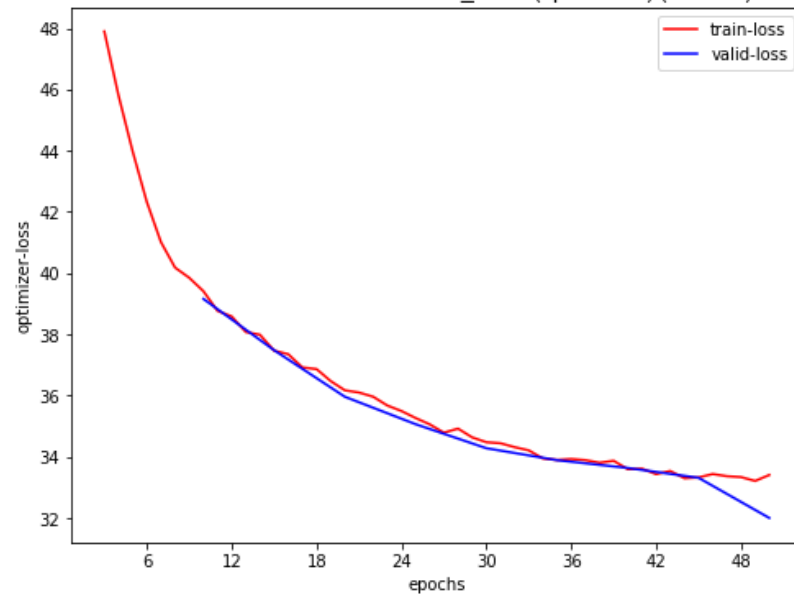
Train/validation loss for model act_sbV1 (epochs 50) ($\nu = 0.95$)



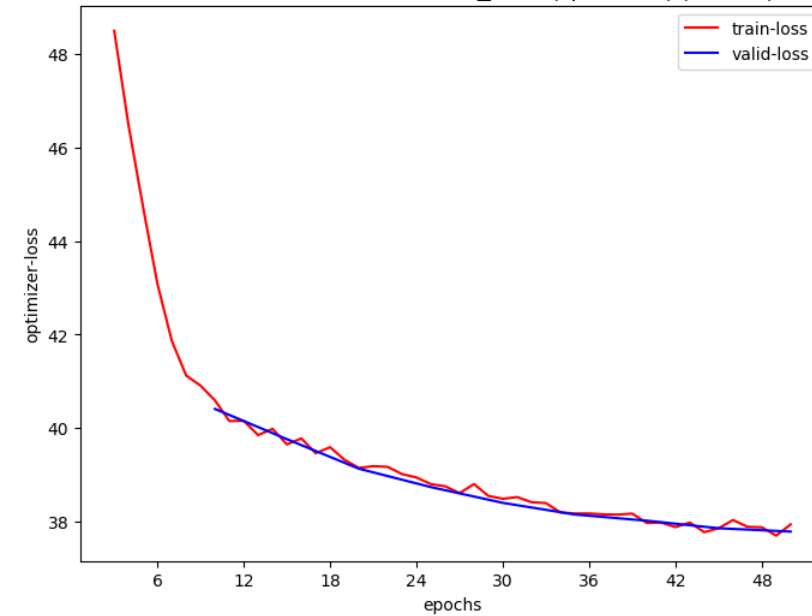
Train/validation loss for model act_sbV1 (epochs 50) ($\nu = 0.9$)



Train/validation loss for model act_sbV1 (epochs 50) ($\nu = 0.7$)



Train/validation loss for model act_sbV1 (epochs 50) ($\nu = 0.3$)

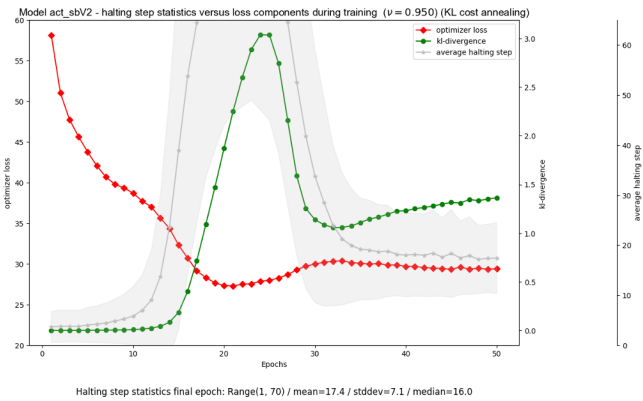
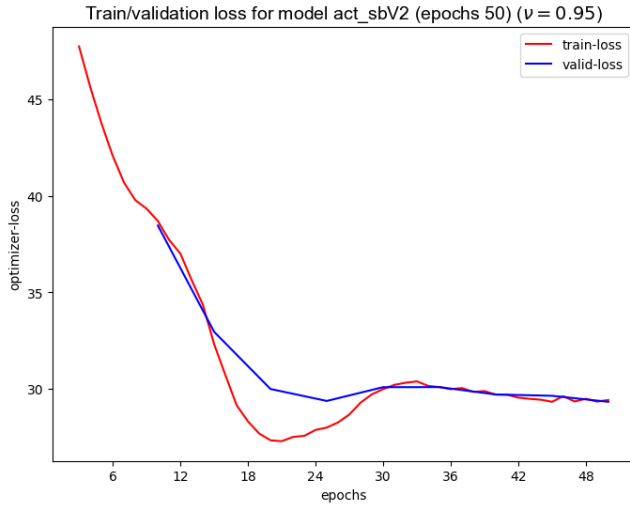


Learning curves ACT-SB models that used KL divergence cost annealing

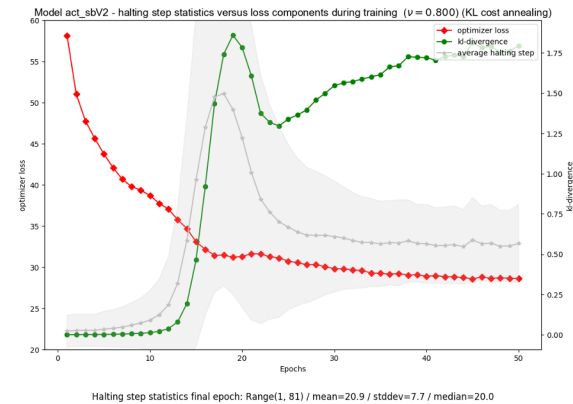
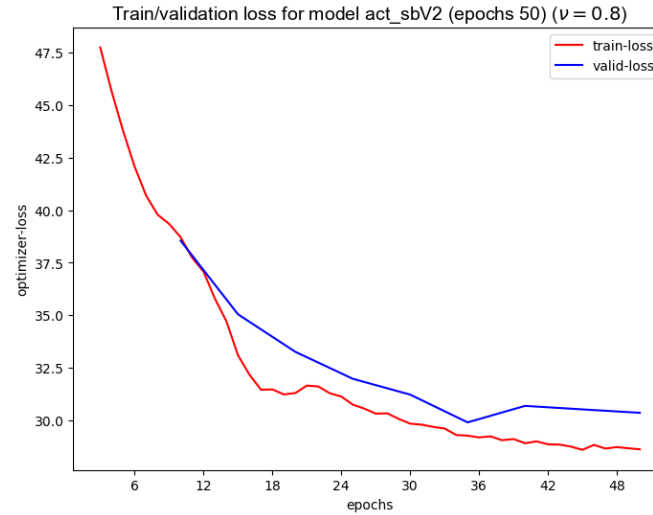
Learning curves of two different act_sb models trained with prior shape parameter values of 0.95 and 0.8.

The KL divergence term of the variational lower bound was scaled according to a *typical* sigmoid annealing schedule (see Bowman et al. 2015 *Generating Sentences from a continuous space*). **Important:** (1) for the 0.95 model the KL weight increased during **40 epochs** up to a value of 1; (2) for the 0.8 model the KL weight increased during the first **20 epochs** up to a value of **0.5**! which was held constant during the remaining 30 epochs.

With KL divergence cost annealing



With KL divergence cost annealing



Without KL divergence cost annealing

