# Appendix for SKIing on Simplices: Kernel Interpolation on the Permutohedral Lattice for Scalable Gaussian Processes

**Sanyam Kapoor** [*1] **Marc Finzi** [*1] **Ke Alexander Wang** [2] **Andrew Gordon Wilson** [1]

## A. Hyperparameters

Table 5 documents all the hyperparameters used for training Simplex-GPs. All kernels use automatic relevance determination (ARD). We find that higher values of $r$ (e.g. 2 or 3) do not meaningfully improve the test RMSE performance, but significantly increase the training time.

*Table 5.* We document all the settings and hyperparameters involved in training Simplex-GPs.

| HYPERPARAMETER | VALUE(S) |
|---|---|
| MAX. EPOCHS | 100 |
| OPTIMIZER | Adam |
| LEARNING RATE | 0.1 |
| CG TRAIN TOLERANCE | 1.0 |
| CG EVAL/TEST TOLERANCE | 0.01 |
| MAX. CG ITERATIONS | 500 |
| CG PRE-CONDITIONER RANK | 100 |
| MAX. LANCZOS ITERATIONS | 100 |
| KERNEL FAMILY | { Matérn-3/2, RBF } |
| BLUR STENCIL ORDER ($r$) | 1 |
| MIN. LIKELIHOOD NOISE ($\sigma^2$) | $\{ 10^{-4}, 10^{-1} \}$ |

## B. Visualizing Training Instabilities

We visualize the training instabilities that arise as a consequence of using a high CG tolerance value. As noted in Section 5.4, we follow the recommendation of Wang et al. (2019), and use a CG tolerance of 1.0 during training and 0.01 during validation and test. We find that the train MLL does not improve monotonically, due to lack of CG convergence, often owing to early truncation. This leads to undesirable behavior in the test RMSE as visualized in Figure 7(a).

As addressed in Section 5.4, a more stable training run is achieved by simply reducing the tolerance to $10^{-4}$, as visualized in Figure 7(b). But this leads to a significant slowdown, defeating the computational gains from Simplex-GPs. Therefore, this remains a noteworthy design decision for practical usage.

## C. Comparing Learned Lengthscales with Exact GPs

When comparing the results from the Simplex-GP approximation to exact GPs via KeOps (Charlier et al., 2020), we find that the learned lengthscales for the Matérn-3/2 ARD kernel agree qualitatively, i.e. the relevance determined by Simplex-GPs corresponds to the relevance determined by KeOps too. In many cases, these agree quantitatively too. This is visualized in Figure 8. The learned scale factors for the kernels are often different, partially accounting for the difference in the magnitude of lengthscales.

This hints that the approximations constructed by Simplex-GPs are meaningful in practice, and similar in quality to exact GPs, than the performance just being coincidental artifact of optimization.

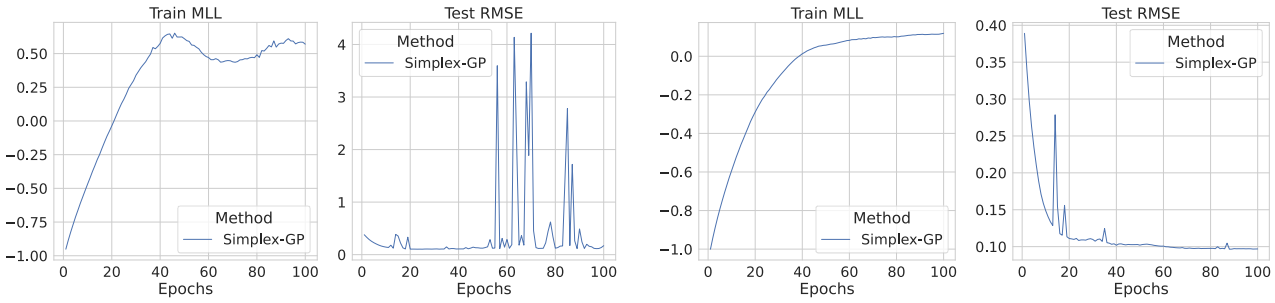


*Figure 7.* We visualize the pathology discussed in Section 5.4, when using conjugate gradients (CG) on the `keggdirected` dataset. We observe similar behavior for other datasets too. (a) Using a high CG error tolerance of 1.0 during training leads to non-monotonic improvements in the train marginal log-likelihood (MLL) due to convergence issues in CG. More significantly, this makes the test RMSE curves look unstable. (b) By simply reducing the CG error tolerance to $10^{-4}$, we are able to stabilize these curves, behaving more favorably.

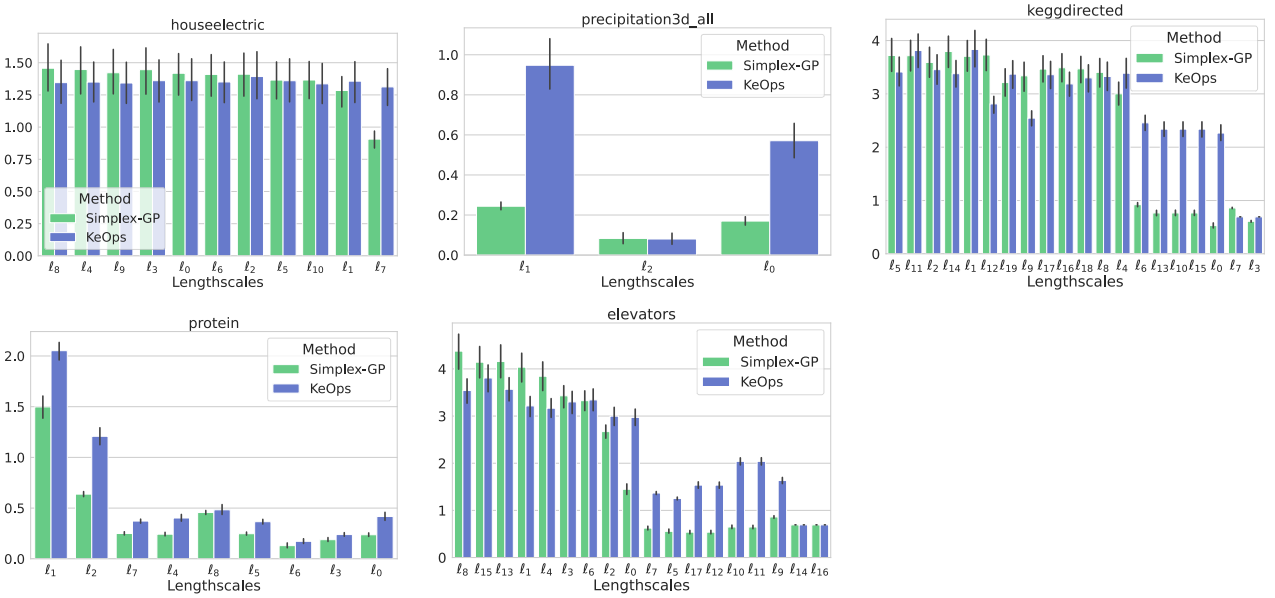


*Figure 8.* For all our benchmark UCI datasets, when comparing the lengthscales between those learned by Simplex-GPs, and those learned by exact GPs using KeOps, we find that the learned values agree in terms of determined relevance. The label $\ell_d$ refers to the lengthscale learned for dimension $d$.