Learning to Superoptimize Programs
Rudy Bunel, Alban Desmaison, M. Pawan Kumar, Philip H.S. Torr, Pushmeet Kohli
Torr Vision Group - OVAL Group, Department of Engineering Science, University of Oxford, UK
Microsoft Research

Introduction
Stochastic Search-based methods [1] can outperform traditional compilers for Program Optimization.

Stochastic Superoptimization [1]
Programs are represented as a series of instructions. An instruction is composed of an opcode and possibly several operands.

Problem Formulation
For a given number of MCMC iterations $T$,
Maximize probability of obtaining good programs at the end.

Learning the proposal distribution
We unroll the execution of the MCMC algorithm, to understand it as a Stochastic Computation Graph [2].

Experimental Protocol
We instrument the STOKE system [1] to be able to estimate the gradients.
We train two types of models:

Results
Data augmentation
Training set and Testing set are obtained by splitting different Hacker’s Delight tasks.
We find equivalent programs with different code from the reference for each task.

Learning the models
Both models can learn and the improvements achieved generalize to unseen tasks.

Using the learned policy
Optimization traces of a uniform proposal distribution against a learned one.

Discussion
We observe that learning the proposal distribution, improves performance. Our method is generic and could be applied to other problems solved using stochastic search.
Further improvements could be achieved by having a richer model to condition over or by also conditioning on the current state of the optimization.