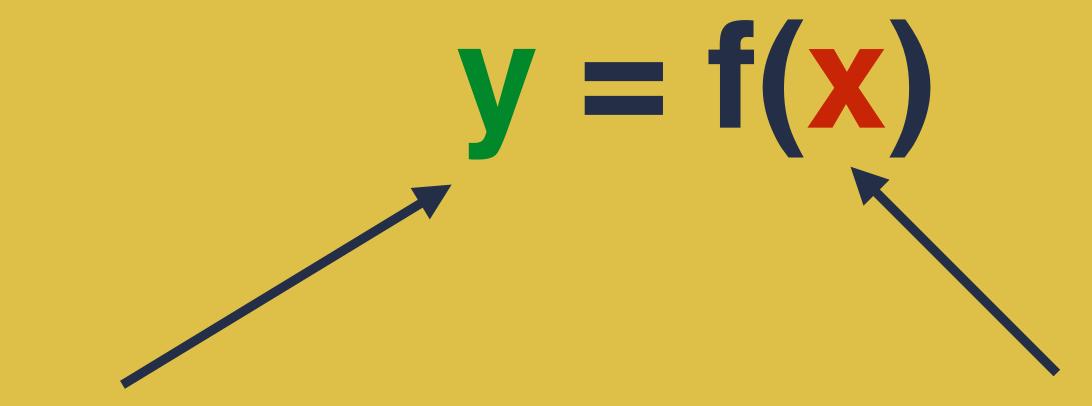
Deep Learning in Compilers

Chris Cummins Pavlos Petoumenos Zheng Wang Hugh Leather





Decisions

CPU or GPU

Workgroup size

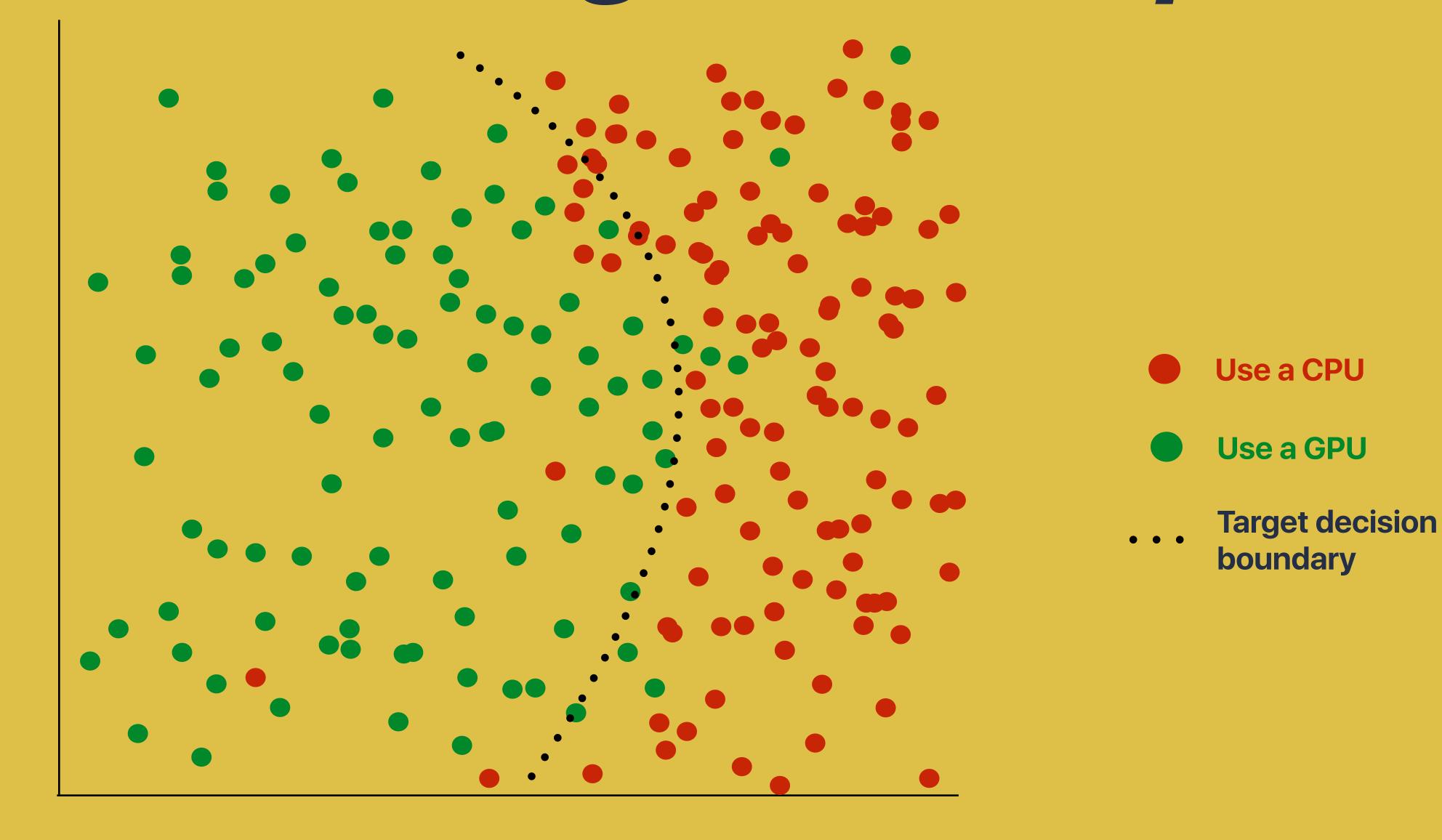
Cflags

Features

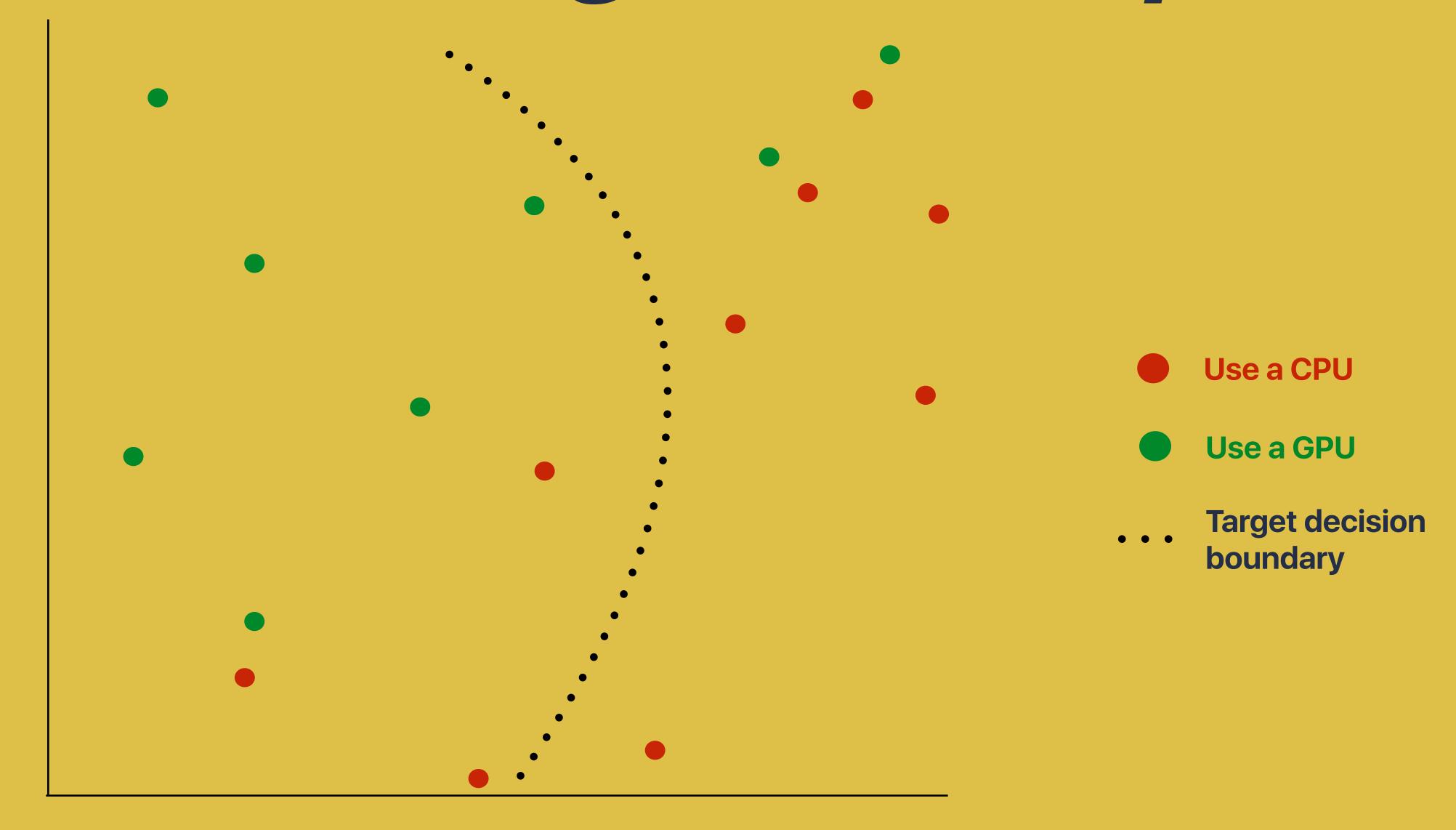
instructions

Arithmetic density

Dataset size

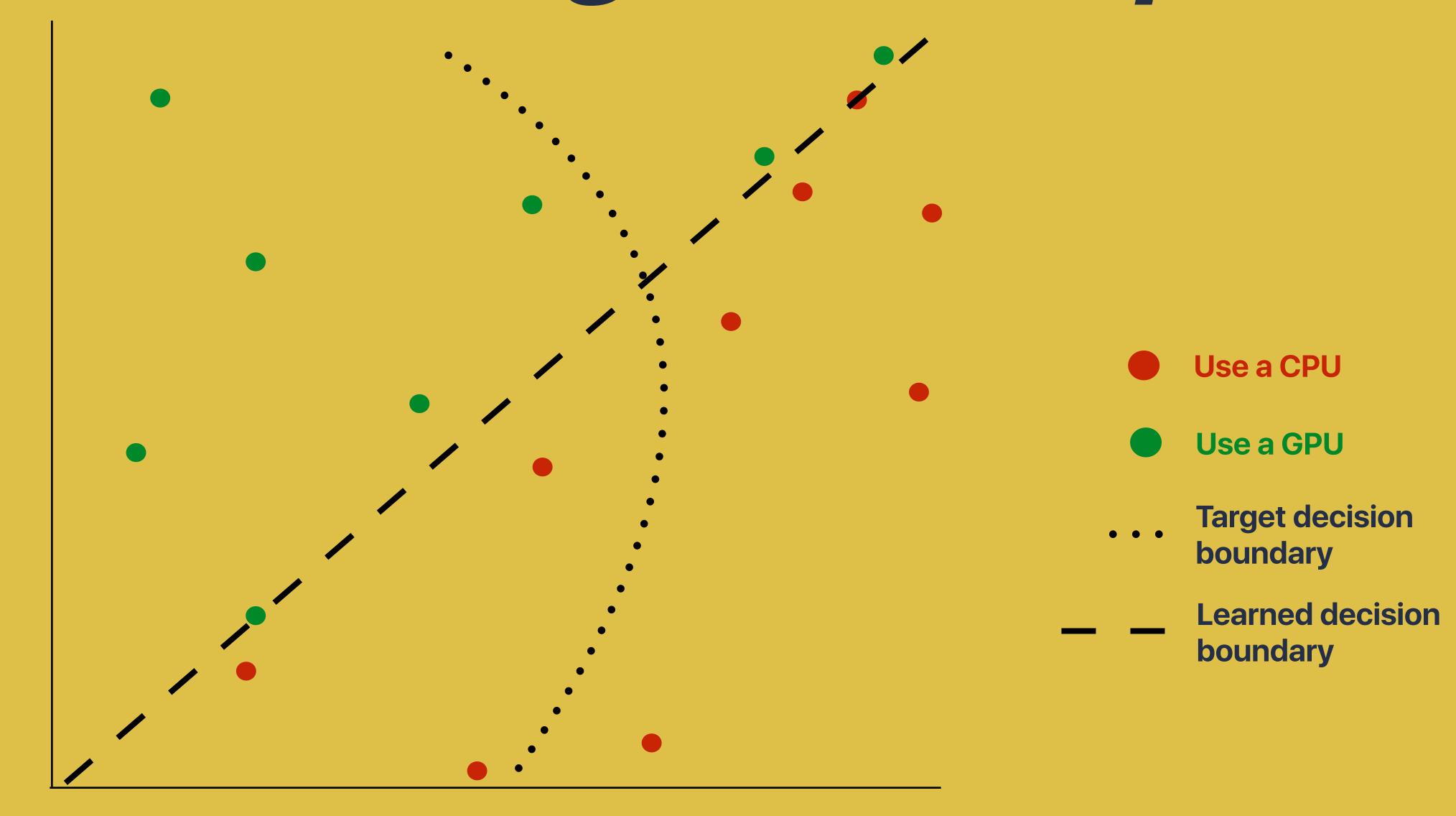


The idea.



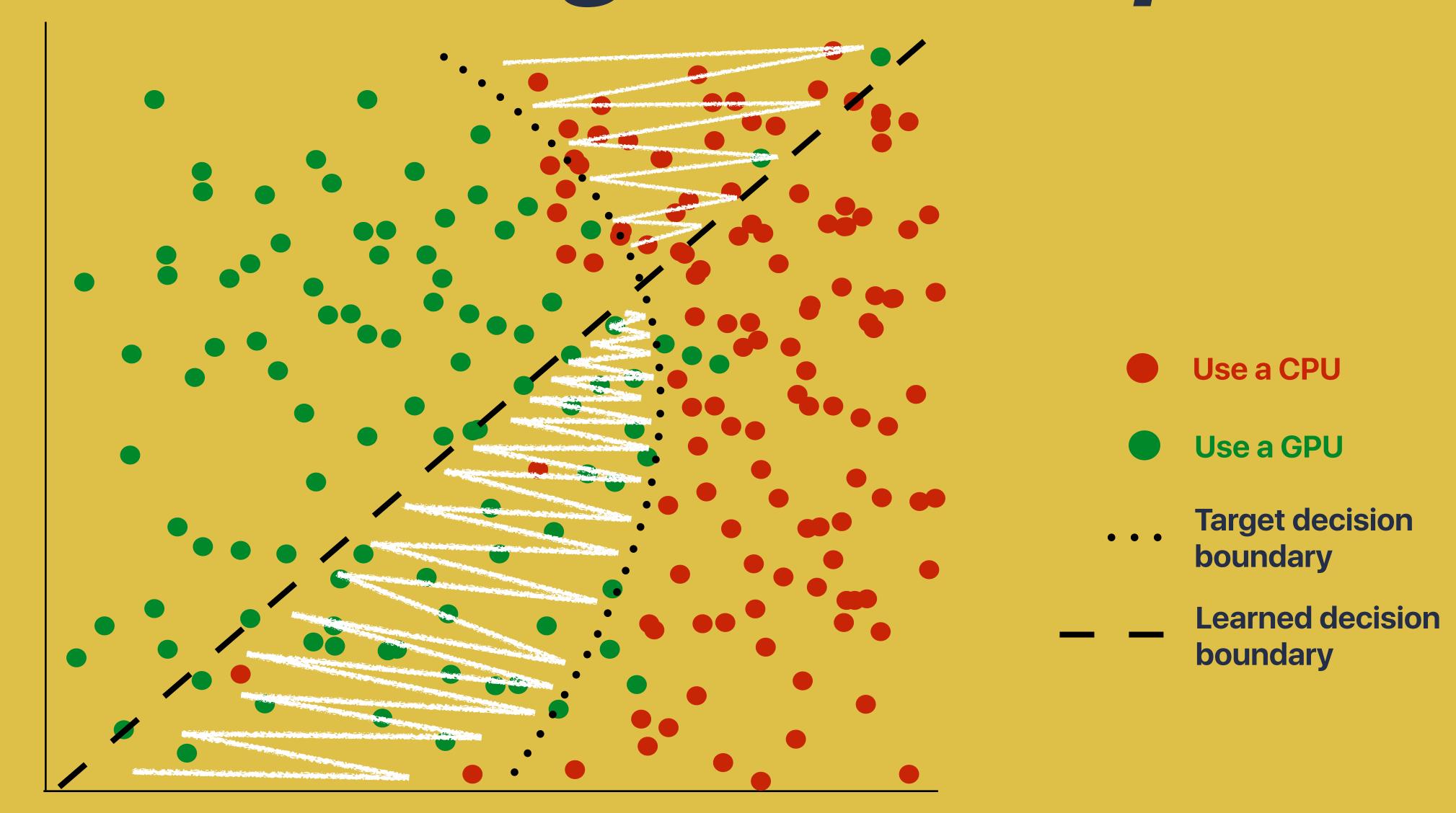
The reality.

*17 benchmarks in avg compiler paper 2013-2016



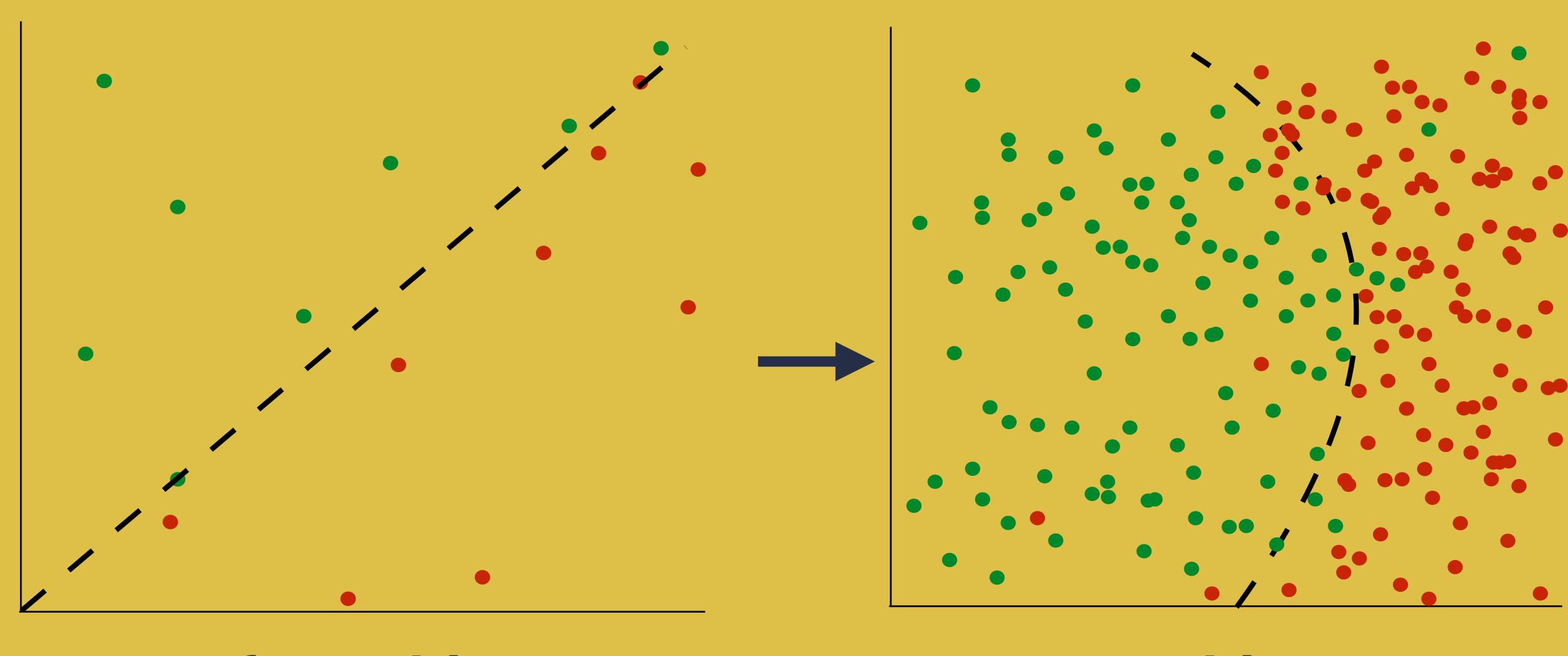
The reality.

*17 benchmarks in avg compiler paper 2013-2016



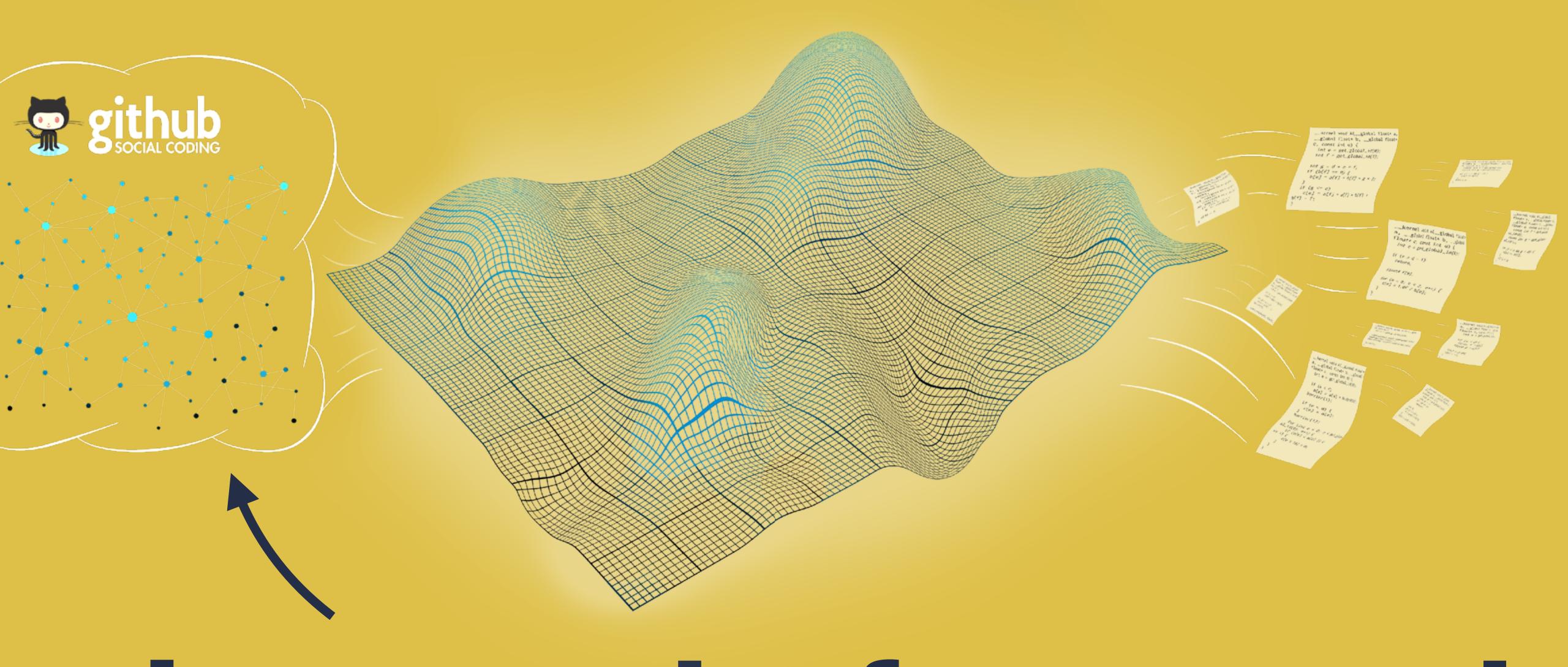
sparse data leads to inaccurate models!

what we need

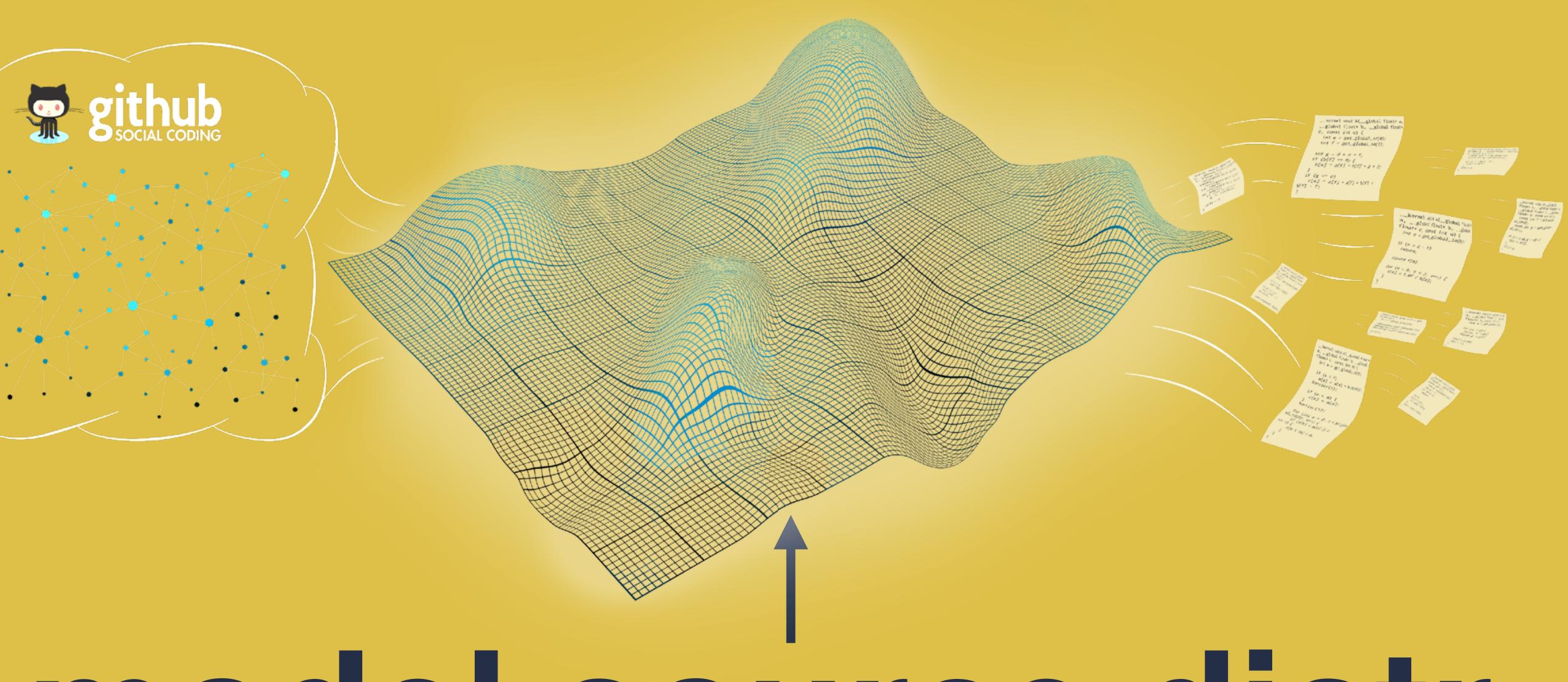


from this

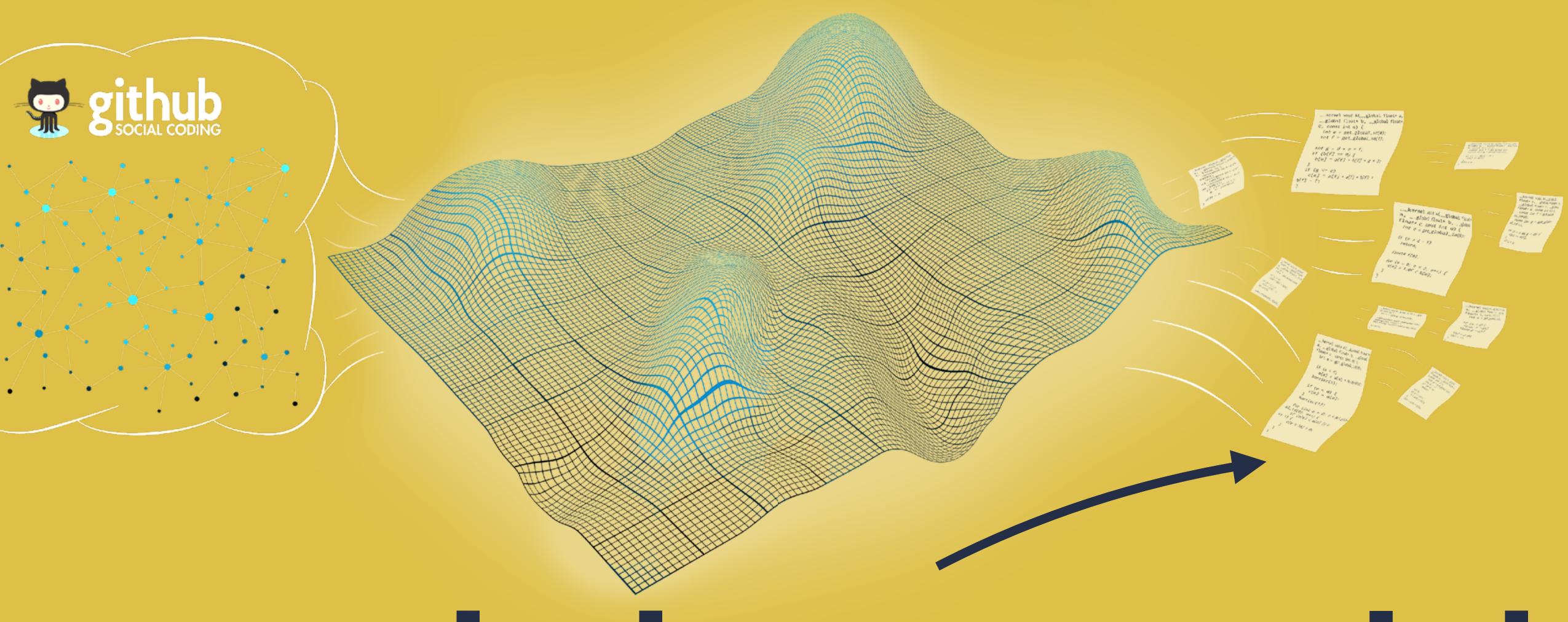
to this



mine code from web



model source distr.



sample lang. model

```
__kernel void A(__global int* a, __global int* b, __global int* c,

→ __global int* d, const uint e) {
      const uint f = get_global_id(0);
      if (e == 0 \&\& f == 0)
        *d - 0:
      else if (f < e) {
        int g = b[f];
        uint h = c[f];
       if (g > 0) {
         a[h] = f;
          h++;
        if (f - e - 1)
14
          *d - h:
15
16
```

Listing 4: Sample 4

```
__kernel void A(__global float* a, __global float*
        int e = get global id(0);
      if (e < d) {
        float f = b[e];
        float g = a[e];
        a[e] - f * 3.141592 f / (f + 1.0 f + e * 1024 - f)  (0.5 f g * 1.0 f)
            \rightarrow / 18.0 f + e / 2.0 f);
      for (e = 0; e < 30; e++) {
10
        c[e] = 0;
11
12
```

13

loat* c,

http://humanorrobot.uk

10

Listing 4: Sample 4

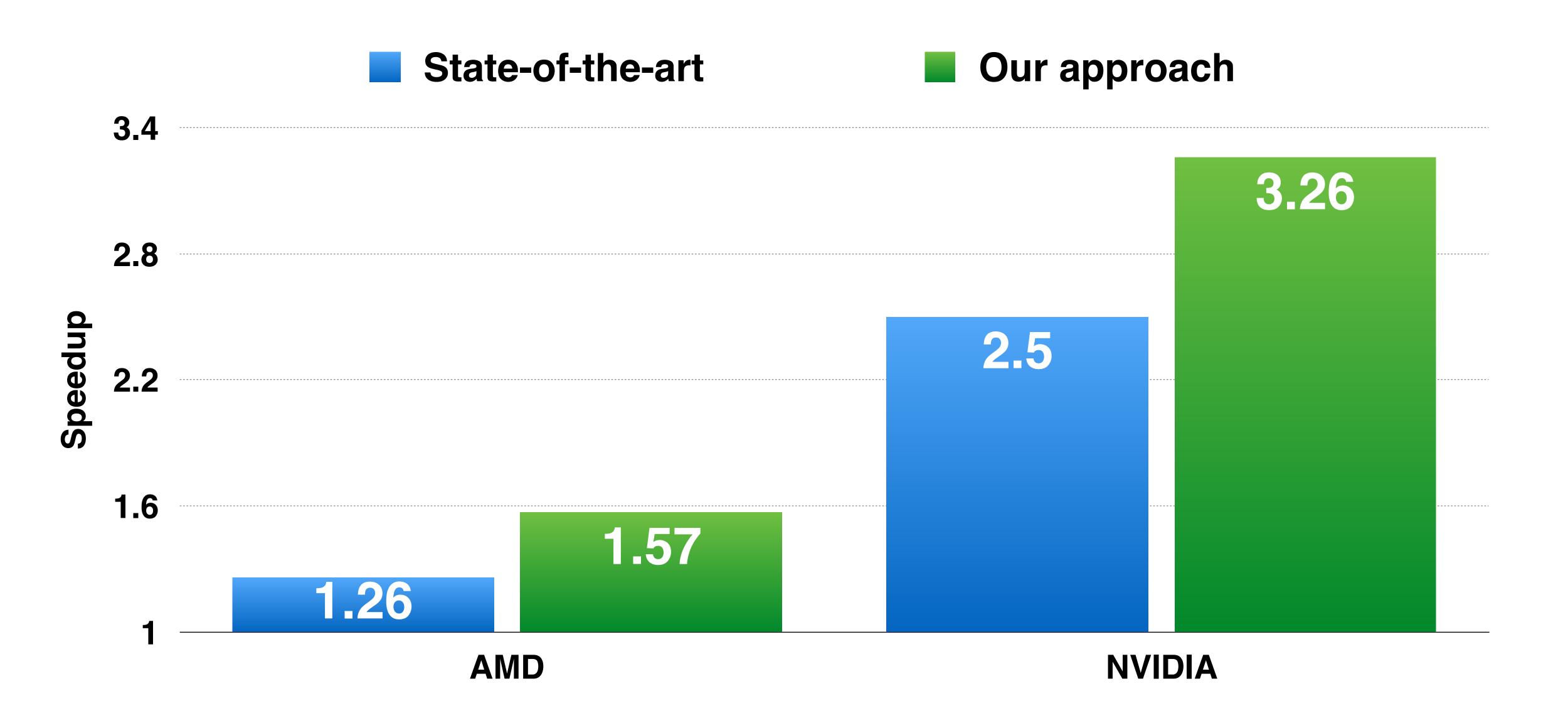
```
kernel void A(int a, \_\_global float*b, \_\_global int*c, \_\_global int*
  \hookrightarrow d, __local int* e, int f) {
int g = get_local_id(0);
e[get\_local\_id(0)] = 0;
barrier (1);
while (g < f) {
  int h = c[g];
  if (h != -1) {
     _{-}global float* i = b + g * a;
    float i = 0:
```

```
__kernel void A(int a, int b, int c, __global const float* d, __global
   const int h = get_local_id(0);
 const int i = get_group_id(0);
 const int j = 4 * i + h;
 const int k = 4 * i + h + a;
  if (4 * i + h + a < c) {
   float l = 0.0;
   float m = 0.0;
   float n = 0.0;
    const float o - d[3 * (4 * i + h + a)];
   const float p = d[3 * (4 * i + h + a) + 1];
   const float q = d[3 * (4 * i + h + a) + 2];
    for (int r = 0; r < c; r++) {
     const float s = d[3 * r] - o;
     const float t = d[3 * r + 1] - p;
             t u = d[3 * r + 2] - q;
             at v = (d[3 * r] - o) * (d[3 * r] - o) + (d[3 * r + 1] - o)
             (d[3 * r + 1] - p) + (d[3 * r + 2] - q) * (d[3 * r + 2] + q)
           float w
                   -e[r] / (((d[3 * r] - o) * (d[3 * r] - o) + (d[3 * r] - o)))
                     (d[3 * r + 1] - p) + (d[3 * r + 2] - q) * (d]
                     [2] - q) + g) * sqrt((d[3 * r] - o) * (d[3 * r] - o)
                     * r + 1] - p) * (d[3 * r + 1] - p) + (d[3 * r + 2]
                  * (d[3 * r + 2] - q) + g));
                  * r | - 0) * w;
     m = m + (a[3 * r + 1] - p) * w;
     n = n + (d[3 * r + 2] - q) * w;
```

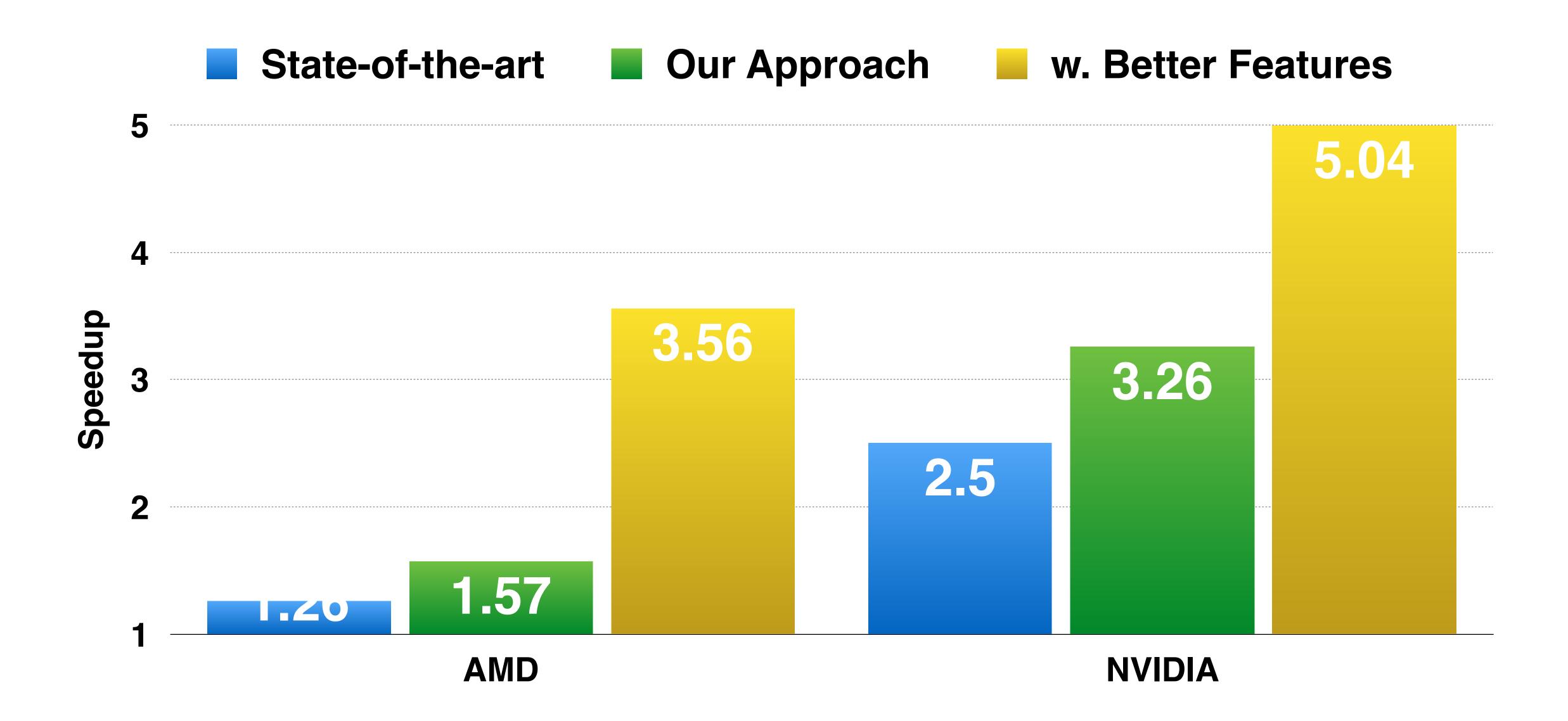
Listing 10: Sample 10

```
__kernel void A(__global ulong *a) {
                                                                  int i, j;
                                                                   struct S0 c_8;
                                                                  struct S0* p_7 - &c_8;
                                                                   struct S0 c_9 = \{
                                                                                                     \{\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x43250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2UL\},\{0x4250E6DL,2U
                                                                                                                  \{0x43250E6DL, 2UL\}, \{0x43250E6DL, 2UL\}, \{0x43250E6DL, 2UL\},
                                                                                                              \{0x43250E6DL, 2UL\}, \{0x43250E6DL, 2UL\}\},
                                                                                                    0x4BF90EDCAD2086BDL,
 10
                                                                  c_8 = c_9;
11
                                                                   barrier (0 | 1);
12
```

7 benchmarks, 1,000 synthetic benchmarks. 1.27x faster



71 benchmarks, 1,000 synthetic benchmarks. 4.30x faster



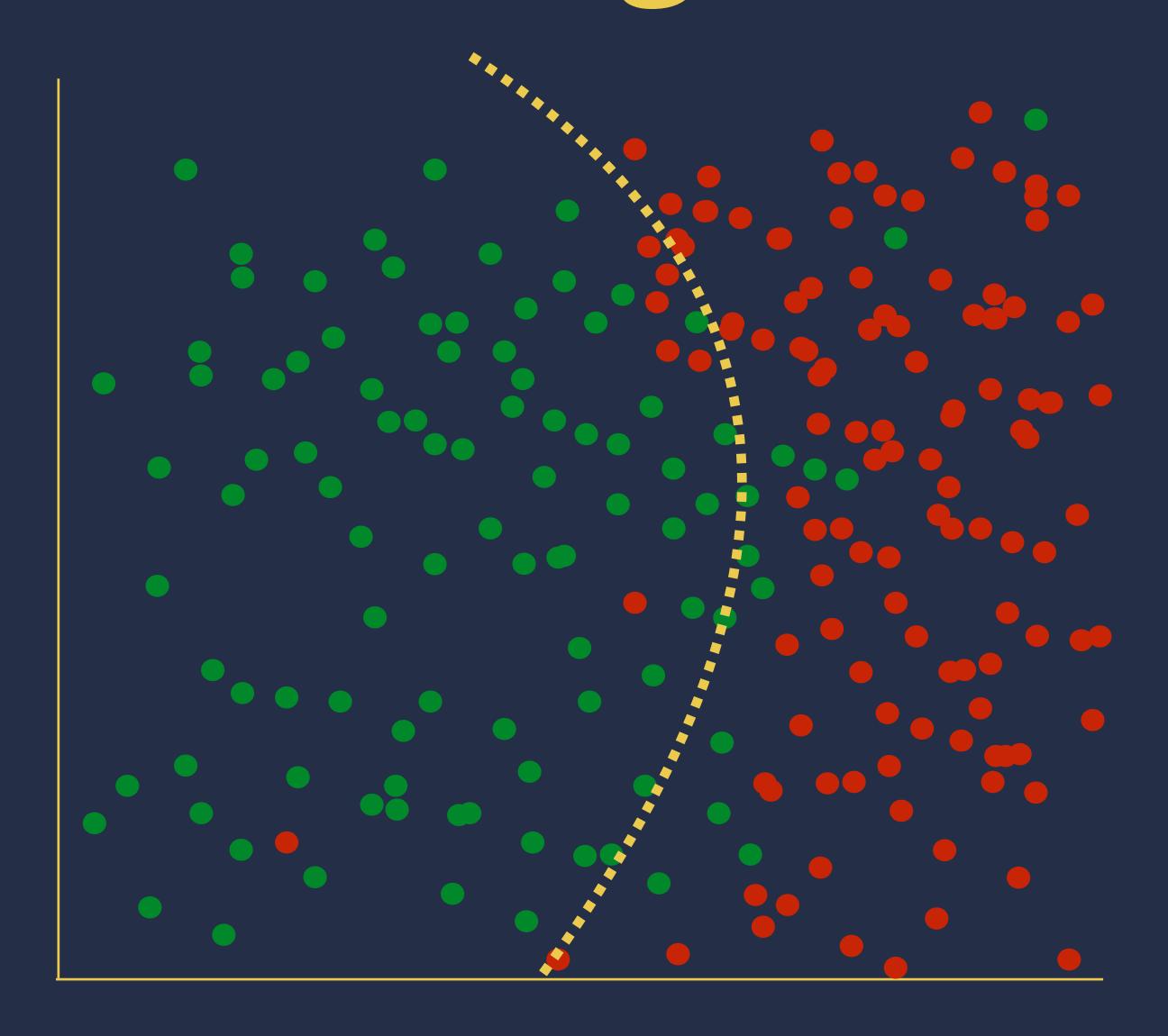
Synthesizing Benchmarks for Predictive Modeling

problem: insufficient benchmarks

first solution for general-purpose benchmark synthesis

turing tested! ;-)

improved model performance and design

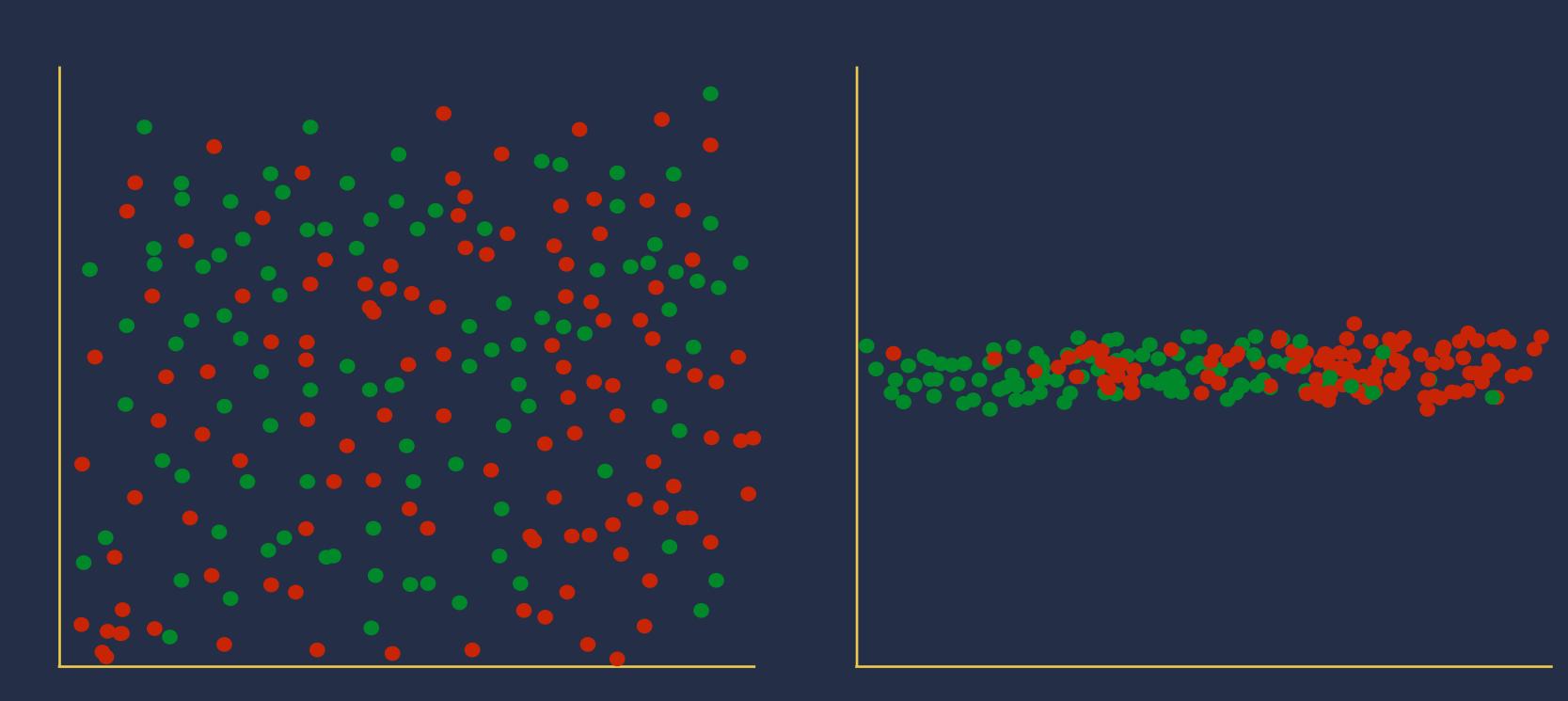


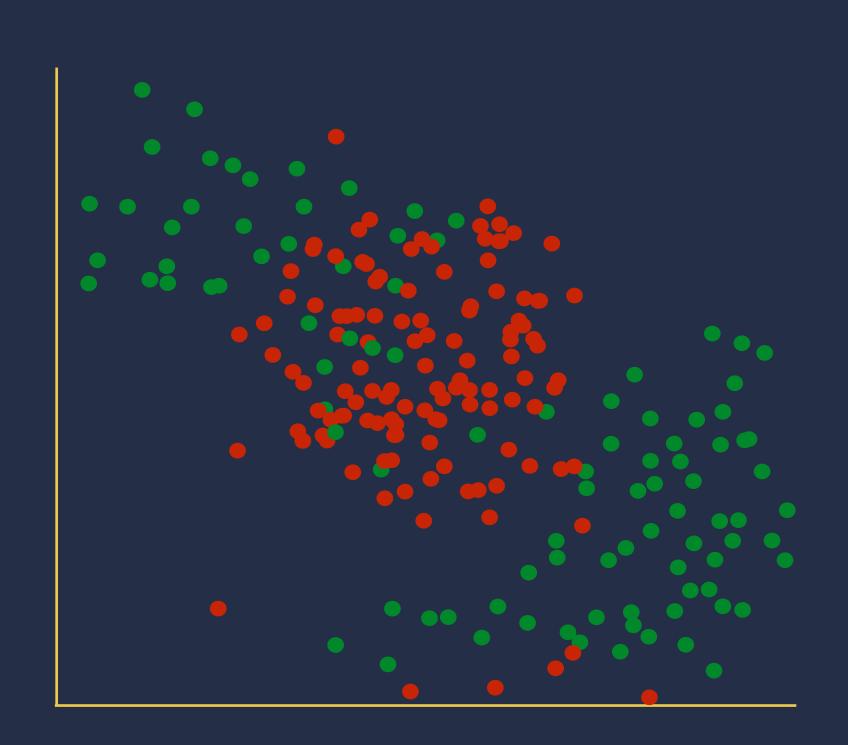
- Use a CPU
- Use a GPU

Target decision boundary

The idea.

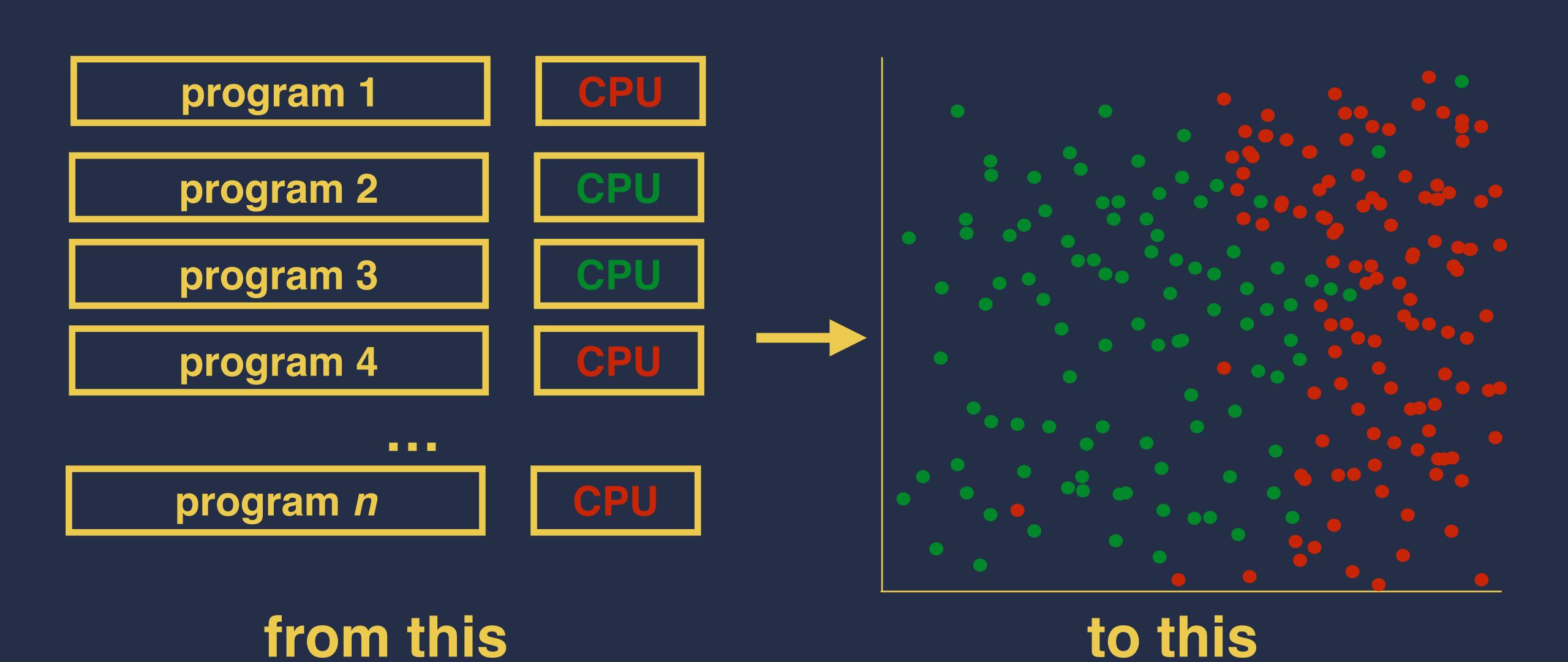
feature design is hard

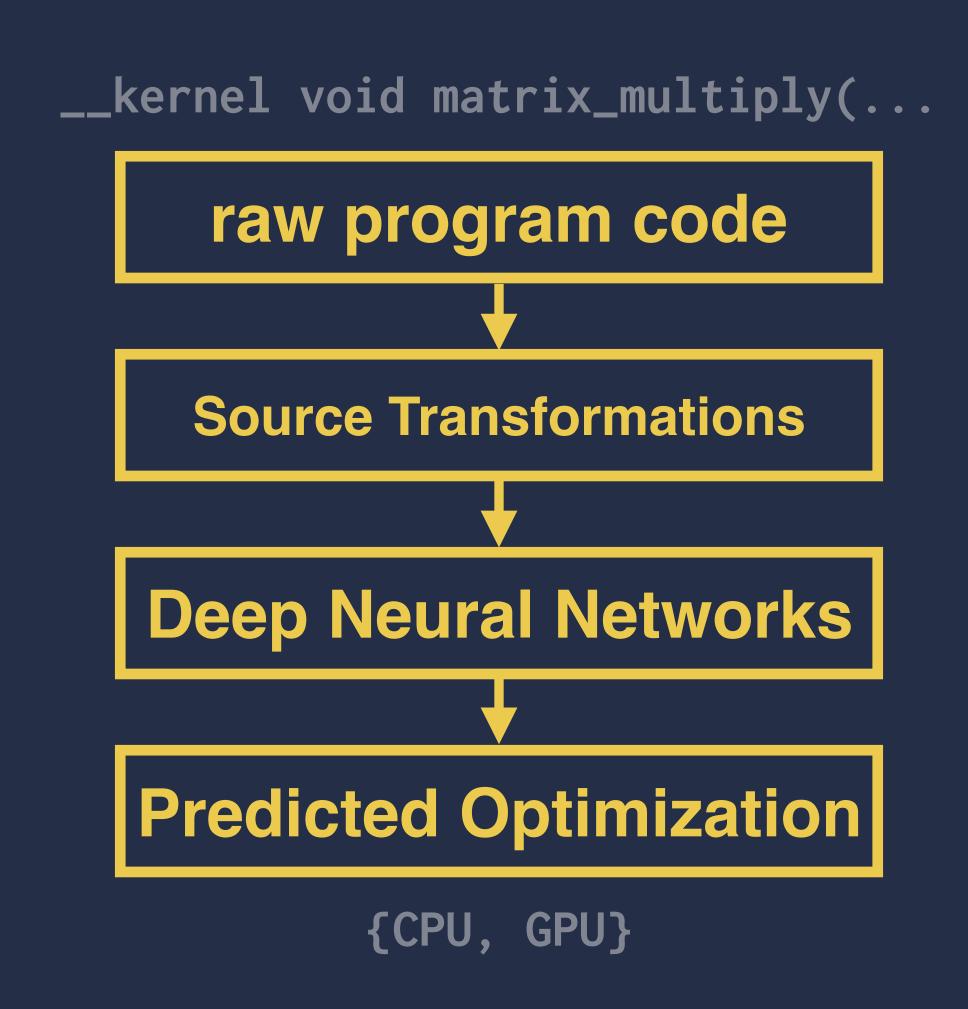


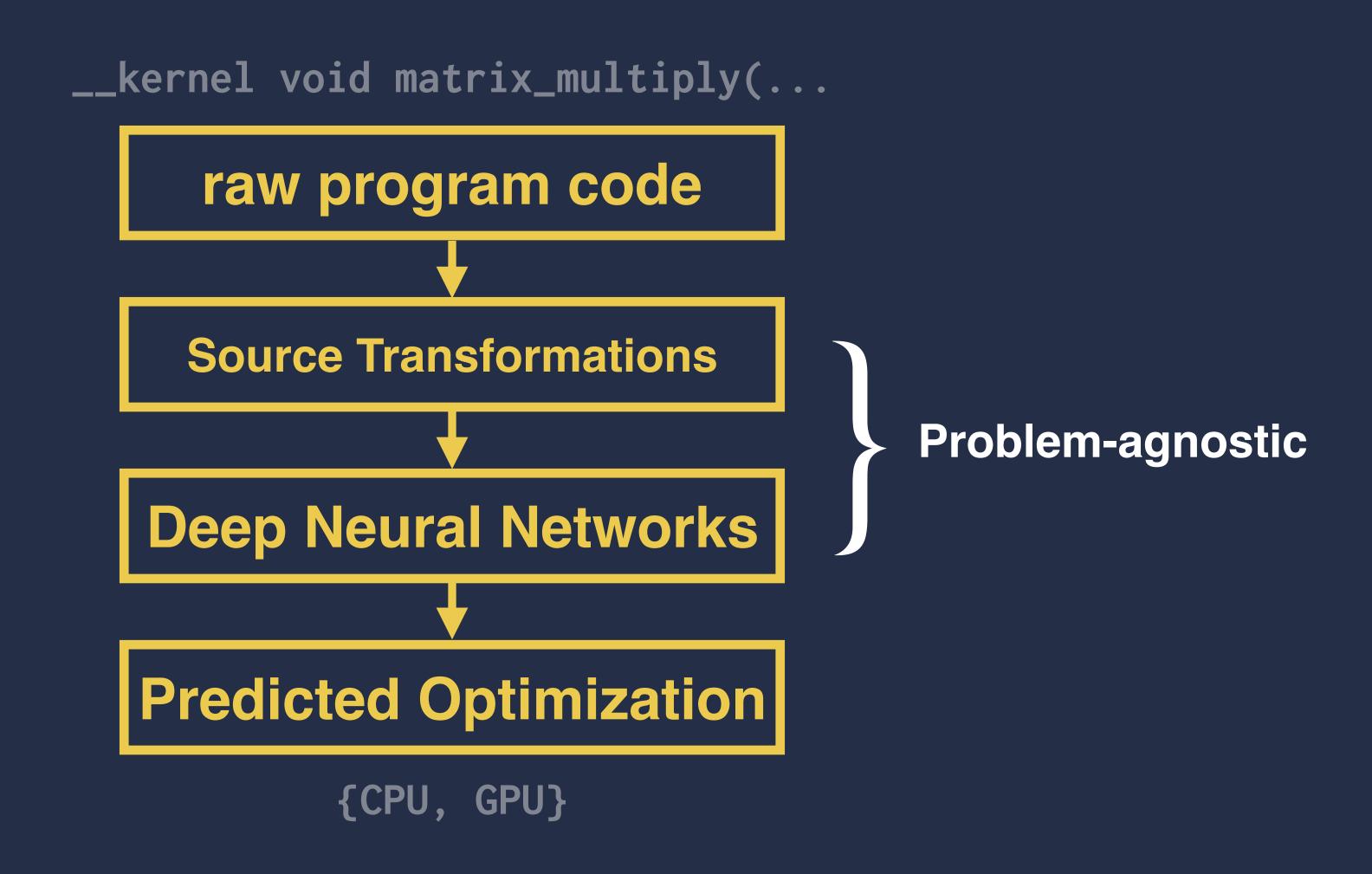


performance depends on good features!

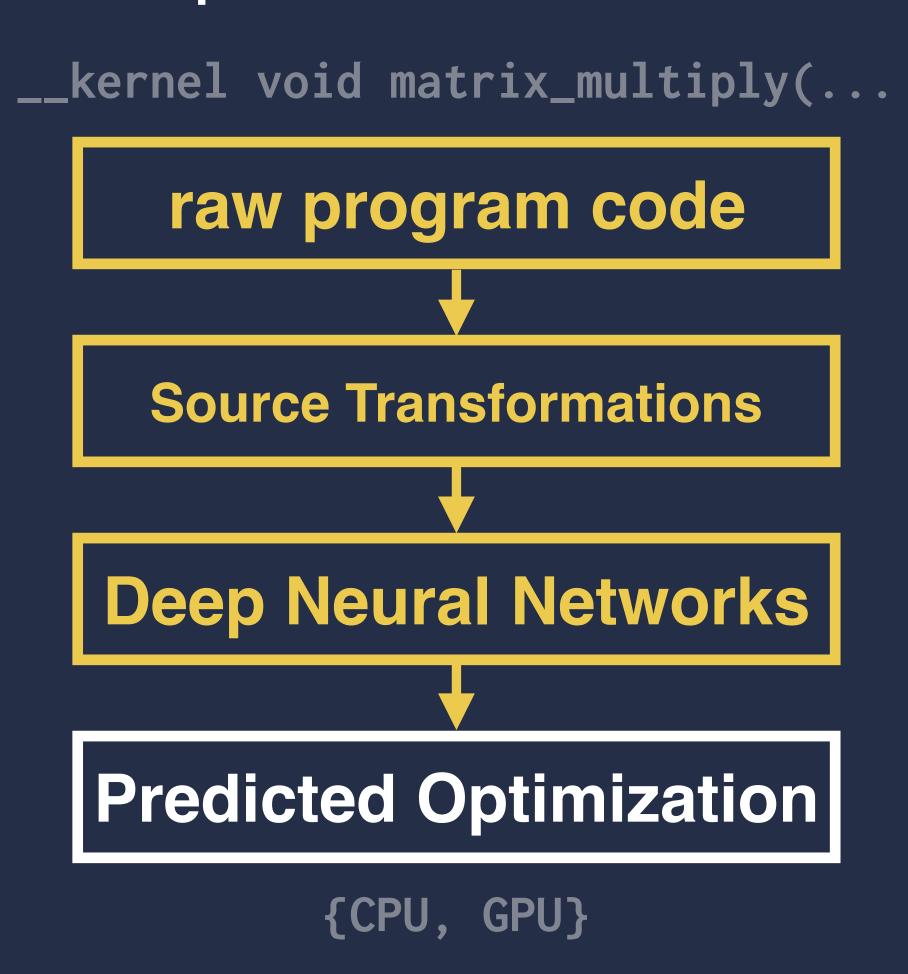
what we need



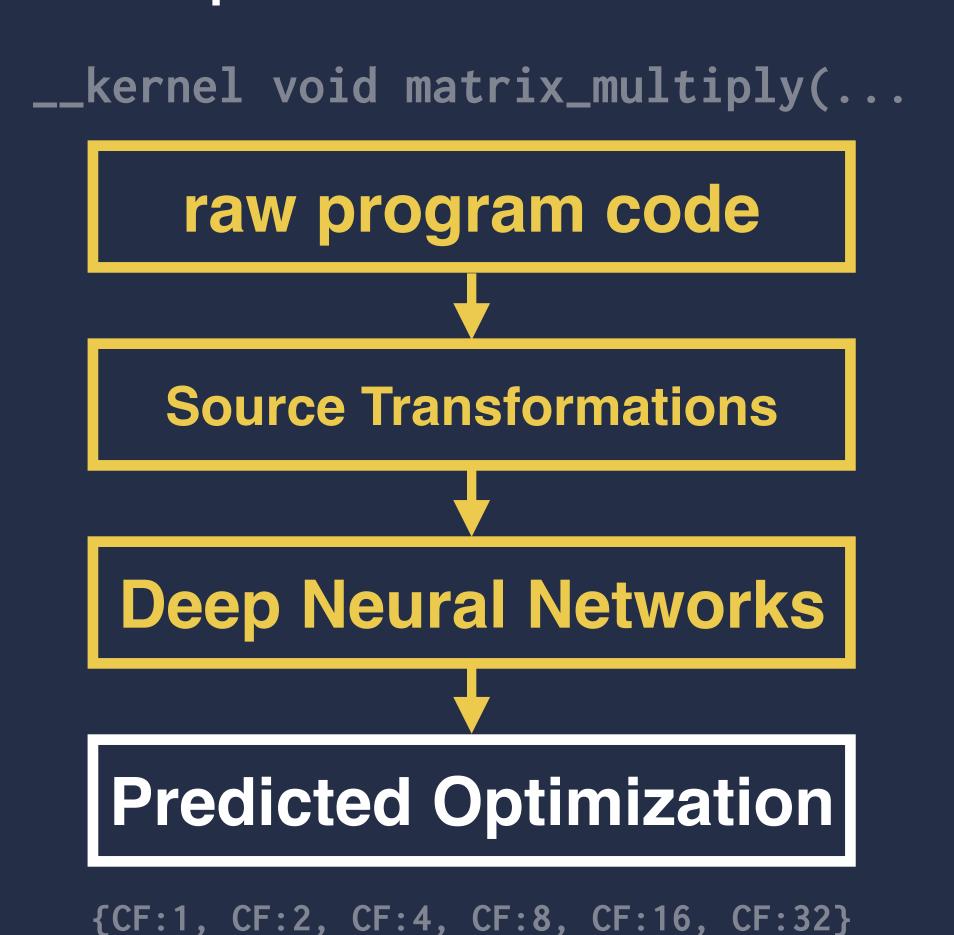




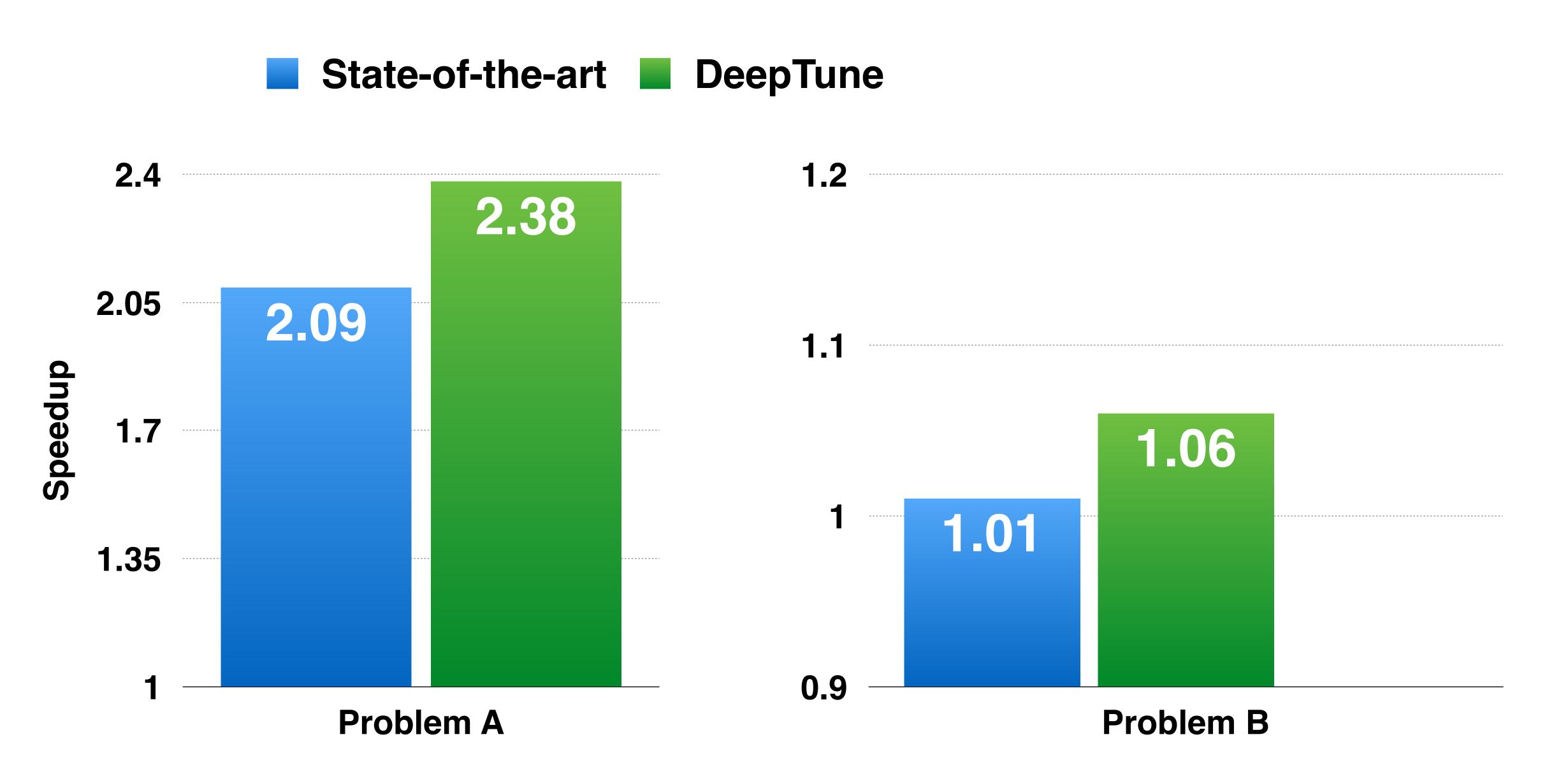
Optimisation Problem A



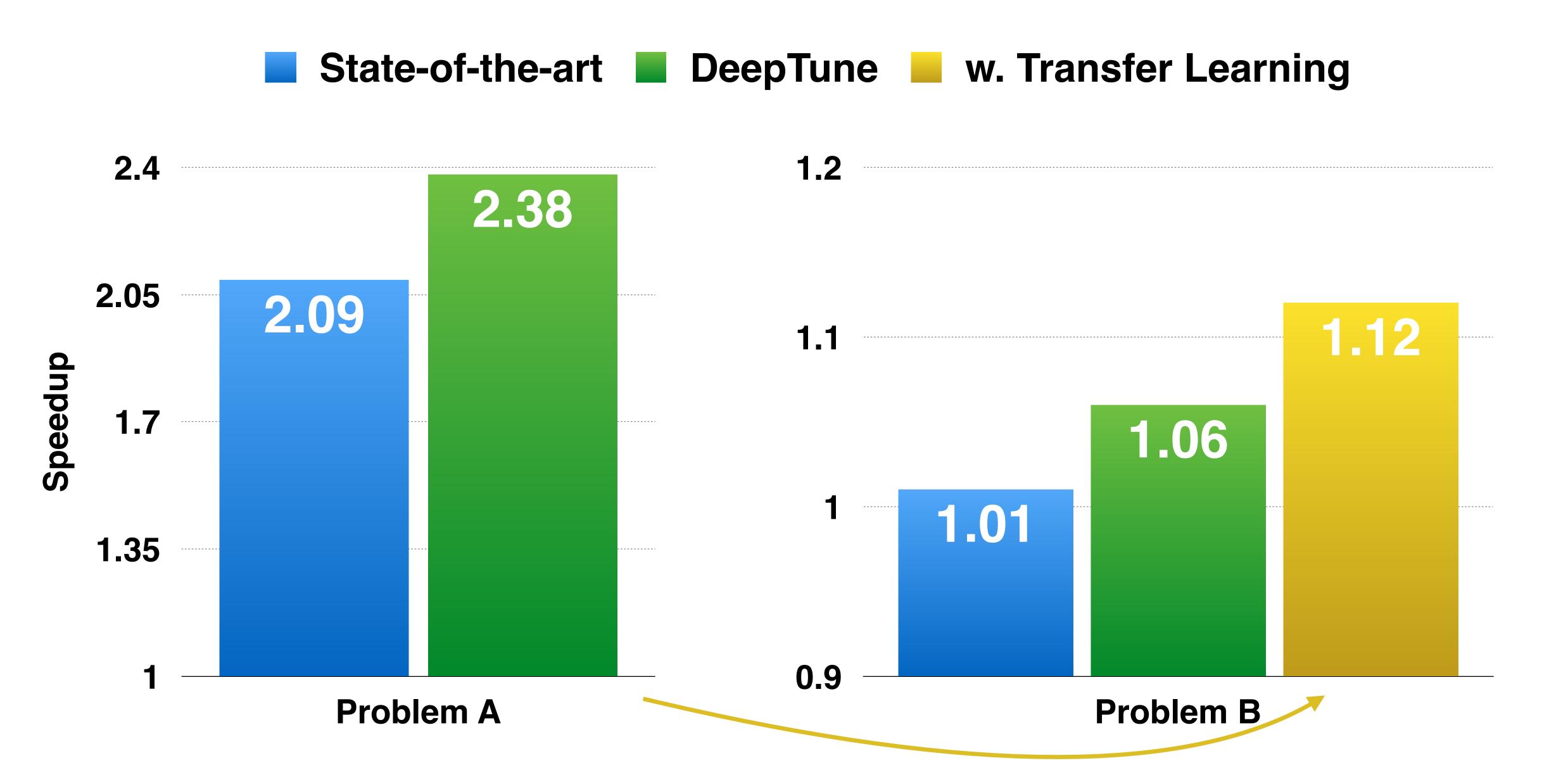
Optimisation Problem B



14% and 12% improvements over state-of-the-art



14% and 12% improvements over state-of-the-art



End-to-end Deep Learning of Optimisation Heuristics

problem: feature design is hard

"featureless" heuristic learning from raw program code

outperforms expert approach

first application of learning *across* optimisation domains

Synthesizing Benchmarks for Predictive Modeling

Further Print menes University of Elinburgh

High Laster

University of Edinburgh Herchar Winfled, soulk

mecrose account amountains tearing and articles account for being complete translate, but force is about good front-bank. Typical matthe beauting appearance in the first of the complete articles of success to match the bank of the complete articles of success to match the bank of the complete. Because of success to match the bank of the complete, the complete articles of success to match the bank of the complete. ver there are typecasy easy a raw after common renormals which was also been successful. is they have very sparse existing data for what are often

real himse denderer vangde in tibilogijkhom generan) — when given the finance form a new answer program, on regram, forefacti-written cale. — peedic good hearbide values for t.

response from hard-weighter bode.

We use our generator for OpenCL programs. (Elegan to summarize by contacting transmitted of programs and should be therefore, even these requires the professions of a same of the entry profession model by 1.22 × 1.5 million, as becomes for the feature of the same of the entry profession model by 1.22 × 1.5 million, as few sources of a same of the entry profession model by 1.22 × 1.5 million, as the feature of the feature of obeing weighter exclusions of the control of the feature of obeing weighter exclusions of the feature of obeing weighter exclusions. The meaning of the medical character of the professional weighter than the same of the training data is a few objects of the feature of the exclusion of the control of the con

and a designative of the contract of the contr

in they have very against mixing data for what are offered to the content of the

Chingeries and Subject Buerigues | D.2.4 | Proposi-cibilities gamenar with embelded accounts actions that de-sign and subject Buerigues | D.2.4 | Proposi-1. Introduction

Archetic modellar is a will research of motion for building operations benefities for will research motion benefities for often record human capters and which the building parent and also product programmed motion are in any

Synthesizing Benchmarks for Predictive Modeling

https://chriscummins.cc/cgo17

EAC & Substitution #9 1- Completous Plant - EAC SWE Lastington:

End-to-end Deep Learning of Optimization Hauristics

Accounts automatic spinning to be unitative as a socretary yer ficting with the completely and discretifying analysis and constant of spinners. Maritims bearings as a proving the compa-jor fearways made four-trains, but the success it is essently to the quality of the features aread. These pleases must be ideal conflictly observed over through a combination of copyer account. pull make directly beginners on the dist and contraction of the

We divides a disposant native Chai secon haviation over no code, painty without using code features. The named net

ration of a parallel program whose decisions must be reads or. To make boudedly a tolkow, or if, a particular optimization should be applied. Med.

jensow, of opposentative groupses. The resident hand the from course code. Our appointent throat that even this constitute between those feature values and frequinitation deviates from one mices performance. The formed events constructed by a section of the constitution of the constitu



strainer, improving the contracts of new strainin, without the bit outputs attending to provide a support of the company attending to provide the company attending to provide the company attending to the company attending attending

End-to-end Deep Learning of Optimisation Heuristics

https://chriscummins.cc/pact17