Distributed training and intelligent simulation for accelerated discovery

Guojing Cong

Oak Ridge National Laboratory

April 11, 2023

Outline

- Distributed learning on modern systems (HPC and cloud) – architecture, network, libraries, algorithms ⟶ the ultimate solver
- Machine learning in scientific simulations – Drug Lead optimization, cancer cell simulation, nanopore materials for CO2 capturing, and thrombosis simulation ⟶ AI for accelerated discovery

Distributed training for deep learning

- “Let data do the programming” calls for big data and big model
  - GPT-3: 175 billion paramters, 570GB, 500 billion tokens, 9 days(*), millions of dollars
  - WuDao: 1.75 trillion parameters, 4.9TB text and Images
  - BaGuaLu: trains up to 174 trillion paramters
- Been-there-Done-That: parallelism, communication, I/O
- Unique to deep learning on converged HPC-AI-cloud systems
  - Convergence
  - Generalization
  - Privacy and Security
- We propose fast algorithms, analyze different approaches, with special focus on scaling, and discuss elastic training in the cloud environment

Distributed training for deep learning

- “Let data do the programming” calls for big data and big model
  - GPT-3: 175 billion paramters, 570GB, 500 billion tokens, 9 days(*), millions of dollars
  - WuDao: 1.75 trillion parameters, 4.9TB text and Images
  - BaGuaLu: trains up to 174 trillion paramters
- Been-there-Done-That: parallelism, communication, I/O
- Unique to deep learning on converged HPC-AI-cloud systems
  - Convergence
  - Generalization
  - Privacy and Security
- We propose fast algorithms, analyze different approaches, with special focus on scaling, and discuss elastic training in the cloud environment

Distributed training for deep learning

- “Let data do the programming” calls for big data and big model
  - GPT-3: 175 billion paramters, 570GB, 500 billion tokens, 9 days(*), millions of dollars
  - WuDao: 1.75 trillion parameters, 4.9TB text and Images
  - BaGuaLu: trains up to 174 trillion paramters
- Been-there-Done-That: parallelism, communication, I/O
- Unique to deep learning on converged HPC-AI-cloud systems
  - Convergence
  - Generalization
  - Privacy and Security
- We propose fast algorithms, analyze different approaches, with special focus on scaling, and discuss elastic training in the cloud environment

Landscape of distributed training approaches

- Asynchronous SGD – downpour, Hogwild!, elastic averaging SGD, and other decentralized methods
- Synchronous SGD – Hardsync (most popular), model averaging (federated learning)
- SGD with other features – quantized gradient, variance reduction, importance sampling, coordinate descent

Problem, notations, and results

- Problem:

$$\min_{\mathbf{w}\in\mathcal{X}} F(\mathbf{w})$$

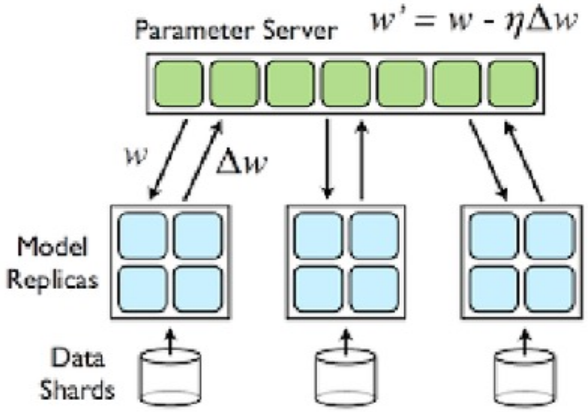
  where F is the objective function, and $F(\mathbf{w}) := \mathbb{E} f(\mathbf{w};\xi)$, *or* $\frac{1}{n}\sum_{j=1}^{n} f_j(\mathbf{w})$

- Results: SGD, $O(1/N)$ convergence with F being twice continuously differentiable and strongly convex, $O(1/\sqrt{N})$ for non-convex; Synchronous SGD, $O(1/\sqrt{NP})$ for non-convex

P	number of processors/learners
K	number of batches processed per each learner between synchronizations
B_n	mini-batch size for n-th update
η_n	step size (learning rate) for n-th update
$\xi^j_{k,s}$	s-th random sample on processor j and step k
$\mathbf{w}$	model weights
μ	momentum
L	Lipschitz constant

Asynchronous stochastic gradient descent (ASGD)

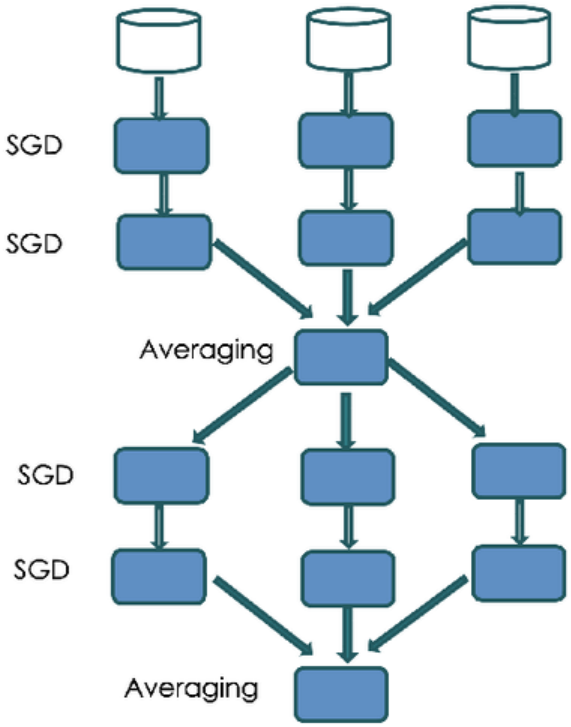
- **Pull:** Get the parameters from the server
- **Compute:** Compute the gradient with respect to randomly selected mini-batch (i.e., a certain number of samples from the dataset)
- **Push and update:** Communicate the gradient to the server. Server then updates the parameters by subtracting this newly communicated gradient multiplied by the learning rate



A K-step averaging algorithm

Algorithm 1 KAVG

```
Initialize $\widetilde{\mathbf{w}}_1$
On $P_j$, $j = 1, \ldots, P$, in parallel :
Learner $P_j$ set $\mathbf{w}^j_1 = \widetilde{\mathbf{w}}_1$
for $n = 1, ..., N$ do
    for $k = 1, ..., K$ do
        randomly sample a mini-batch of size $B_n$ and update:
        $$\mathbf{w}^j_{n+k} \leftarrow \mathbf{w}^j_{n+k-1} - \frac{\eta_n}{B_n} \sum_{s=1}^{B_n} \nabla F(\mathbf{w}^j_{n+k-1}; \xi^j_{k,s})$$
    end for
    Synchronize $\widetilde{\mathbf{w}}_{n+1} = \frac{1}{P} \sum_{j=1}^{P} \mathbf{w}^j_{n+K}$
end for
```



Best Distrubuted Solver?

Practically, for the same data samples processed:

- Which scales with P?
- Which progresses faster towards local optima?
- Which has lower communicaton cost?

We investigate

$$\frac{1}{N}\mathbb{E}\sum_{n=1}^{N}\|\nabla F(\widetilde{\mathbf{w}}_n)\|_2^2,$$ AKA, convergence guarantee

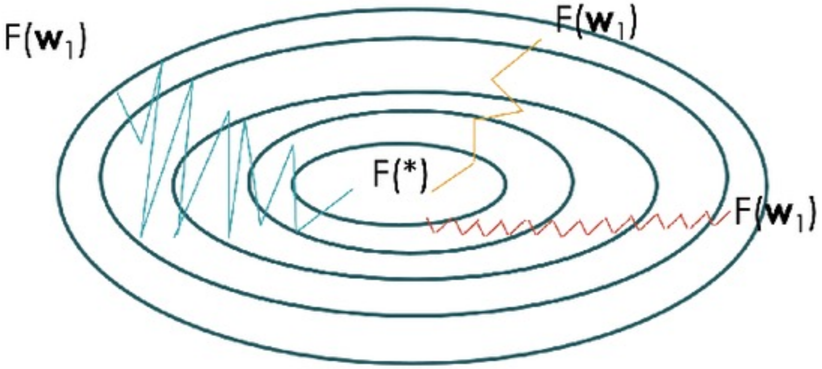
Best Distrubuted Solver?

Practically, for the same data samples processed:

- Which scales with P?
- Which progresses faster towards local optima?
- Which has lower communicaton cost?

We investigate

$$\frac{1}{N}\mathbb{E}\sum_{n=1}^{N}\|\nabla F(\widetilde{\mathbf{w}}_n)\|_2^2,$$ AKA, convergence guarantee



KAVG scales better than ASGD

For ASGD, with fixed stepsize

$$\boxed{\frac{1}{N}\mathbb{E}\sum_{n=1}^{N}\|\nabla F(\widetilde{\mathbf{w}}_n)\|_2^2 \leq \Big[\frac{C_0(F(\widetilde{\mathbf{w}}_1) - F^*)}{N\bar{\eta}} + \frac{C_1 L^2 \bar{\eta}^2 M^2 P}{2\bar{B}}\Big]}$$

where C_0 and C_1 are constants independent of P

For KAVG, with fixed stepsize

$$\boxed{\frac{1}{N}\mathbb{E}\sum_{n=1}^{N}\|\nabla F(\widetilde{\mathbf{w}}_n)\|_2^2 \leq \Big[\frac{2(F(\widetilde{\mathbf{w}}_1) - F^*)}{N(K-1+\delta)\bar{\eta}} + \frac{LK\bar{\eta}M}{\bar{B}(K-1+\delta)}\Big(\frac{K}{P} + \frac{L(2K-1)(K-1)\bar{\eta}}{6}\Big)\Big]}$$

where $0 < \delta < 1$

KAVG allows for larger stepsize

- Stepsize schedule for ASGD:

$$\sum_{n=1}^{\infty} \eta_n = \infty; \boxed{\sum_{n=1}^{\infty} \eta_n^2 < \infty}. \ (e.g. \eta_n = \Theta\left(\frac{1}{n^p}\right), \ \frac{1}{2} < p \leq 1).$$

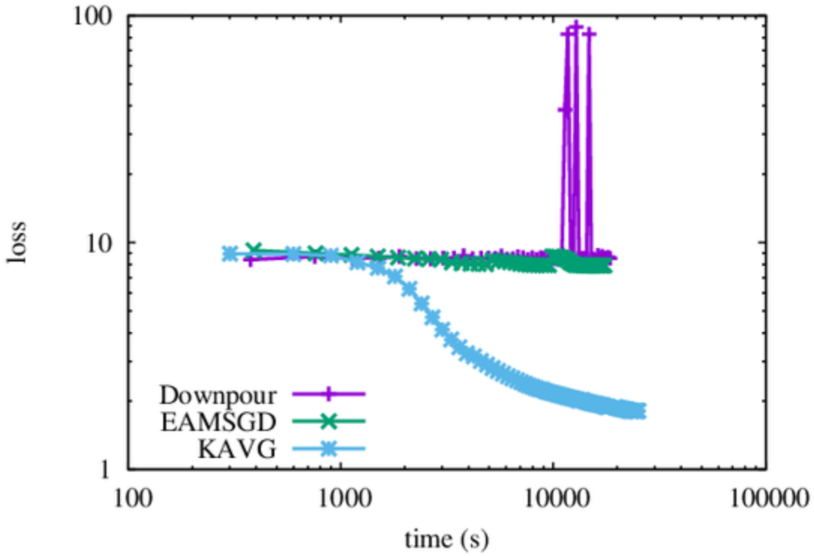
- Stepsize schedule for KAVG:

$$\sum_{n=1}^{\infty} \eta_n * \left(\frac{PL\eta_n(K-1)+1}{PL\eta_n K+1}\right) = \infty; \boxed{\sum_{n=1}^{\infty} \eta_n^3 < \infty}, \ or \ \sum_{n=1}^{\infty} \frac{\eta_n^2}{B_j P} < \infty.$$

$$\eta_n = \begin{cases} \Theta(\frac{1}{n^p}), \ \frac{1}{3} < p \leq 1, \ if \ \sum_{n=1}^{\infty} \eta_n^3 < \infty; \\ \Theta(\frac{\sqrt{B_n P}}{n^p}), \ \frac{1}{2} < p \leq 1, \ if \ \sum_{n=1}^{\infty} \frac{\eta_n^2}{B_j P} < \infty. \end{cases}$$

A real-world example: speech recognition

- The problem: acoustic modeling using hybrid HMM/NN. One“frame” per 10 ms., with 94M frames for the 260-hour Switchboard American English telephone conversational task, and 708M frames from the 2000-hour dataset; 32,000 HMM states
- The NN: a 4-layer bidirectional LSTM with a window of 21 frames. 512 units per direction per layer
- Notoriously hard to scale



20 learners (log — log plot). All use learning rate 0.01

Optimal K

If communication is free, do we want frequent communication?

Let $S = N * K$ be a constant. Suppose that KAVG is run with a fixed stepsize $\eta_n = \bar{\eta}$, and a fixed batch size $B_n = \bar{B}$ for all $n \in \mathbb{N}$ satisfying

$$\frac{LK\bar{\eta}}{2} \geq \frac{1}{P}, \textit{ and } \bar{B} \geq \frac{3L^2K\bar{\eta}^2 M_G}{2}.$$

If

$$\frac{(1-\delta)(F(\widetilde{\mathbf{w}}_1) - F^*)}{S\bar{\eta}\delta} > \frac{(3\delta - 1)L\bar{\eta}M}{2\delta P\bar{B}} + \frac{L^2\bar{\eta}^2 M}{3\bar{B}}$$

the optimal choice of K is greater than 1

Optimal K

If communication is free, do we want frequent communication?

Let $S = N * K$ be a constant. Suppose that KAVG is run with a fixed stepsize $\eta_n = \bar{\eta}$, and a fixed batch size $B_n = \bar{B}$ for all $n \in \mathbb{N}$ satisfying

$$\frac{LK\bar{\eta}}{2} \geq \frac{1}{P}, \text{ and } \bar{B} \geq \frac{3L^2K\bar{\eta}^2M_G}{2}.$$

If

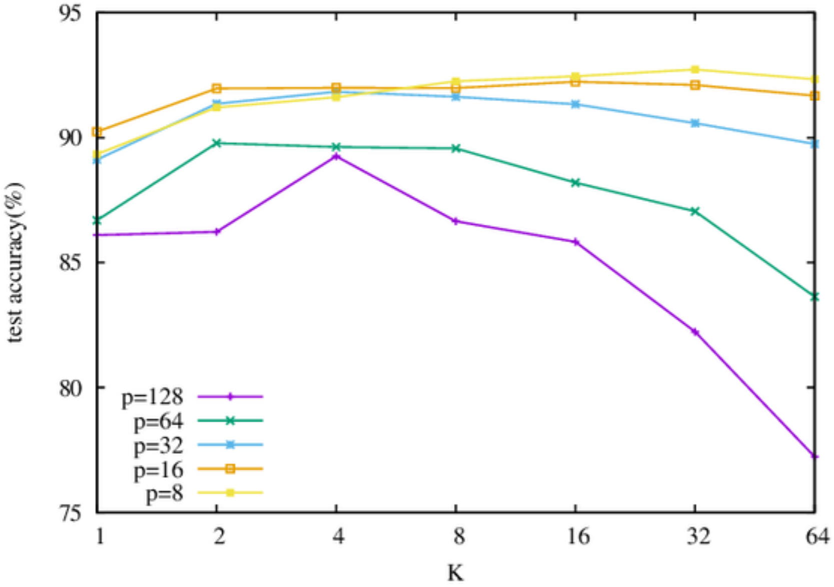
$$\frac{(1-\delta)(F(\widetilde{\mathbf{w}}_1) - F^*)}{S\bar{\eta}\delta} > \frac{(3\delta - 1)L\bar{\eta}M}{2\delta P\bar{B}} + \frac{L^2\bar{\eta}^2M}{3\bar{B}}$$

the optimal choice of K is greater than 1

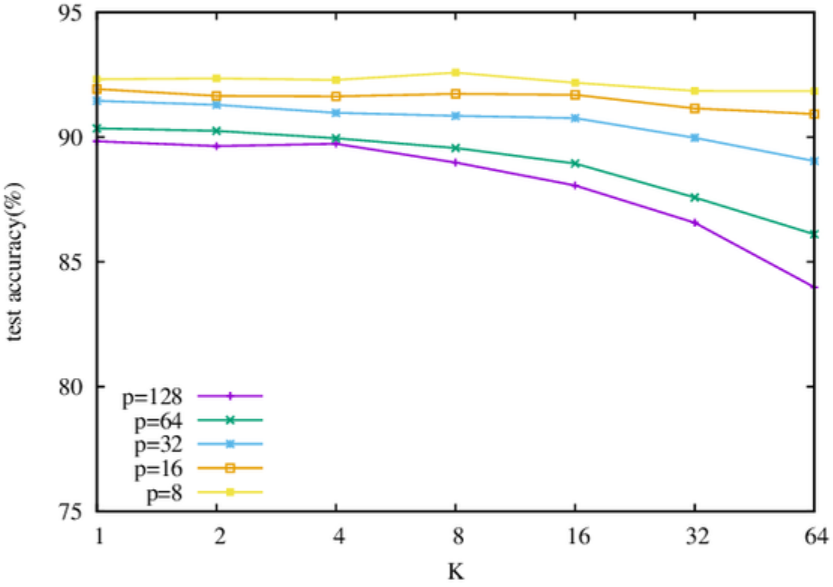
K can be very large

Experiments with CIFAR-10 with up to 128 learners

VGG



NIN

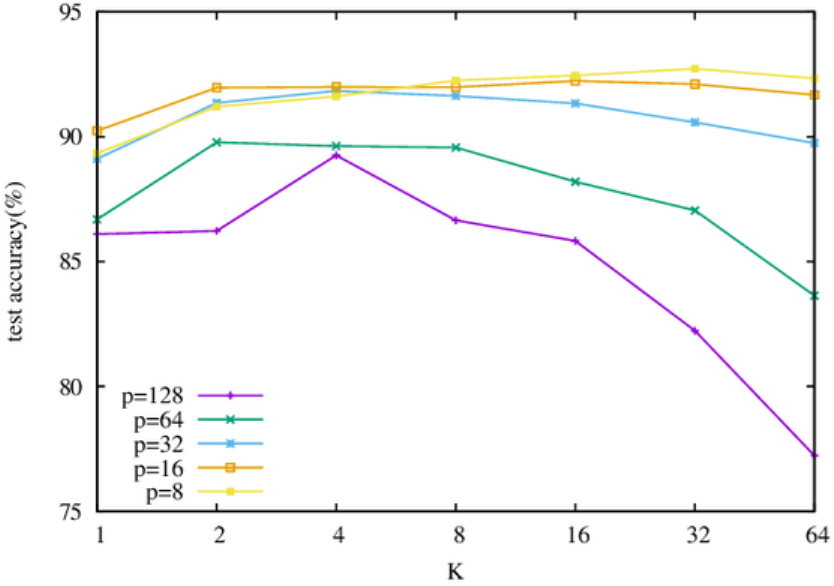


With *ResNet-18*, K_{opt} can be so large that only 1 synchronization per epoch is needed

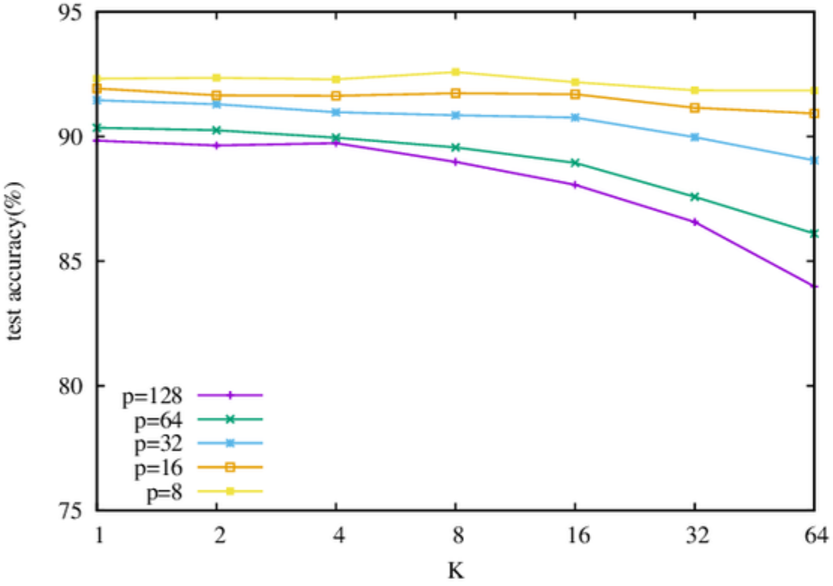
K can be very large

Experiments with CIFAR-10 with up to 128 learners

VGG



NIN



With *ResNet-18*, K_{opt} can be so large that only 1 synchronization per epoch is needed

So communications do not have to be too frequent...

Do we just increase P for fast convergence?

Convergence challenge for large P:

Let $S = NPBK$ be constant, then

$$\frac{1}{N}\mathbb{E}\sum_{n=1}^{N}\|\nabla F(\widetilde{\mathbf{w}}_n)\|_2^2 \le \boxed{\Big[\frac{2(F(\widetilde{\mathbf{w}}_1)-F^*)PBK}{S(K-1+\delta)\bar{\eta}}} + \frac{LK\bar{\eta}M}{\bar{B}(K-1+\delta)}\Big(\frac{K}{P}+\frac{L(2K-1)(K-1)\bar{\eta}}{6}\Big)\Big],$$

Increasing P increases the first term and hence the bound on convergence guarantee

So communications do not have to be too frequent...

Do we just increase P for fast convergence?
Convergence challenge for large P:
Let $S = NPBK$ be constant, then

$$\frac{1}{N}\mathbb{E}\sum_{n=1}^{N}\|\nabla F(\widetilde{\mathbf{w}}_n)\|_2^2$$
$$\leq \boxed{\Big[\frac{2(F(\widetilde{\mathbf{w}}_1) - F^*)PBK}{S(K-1+\delta)\bar{\eta}}} + \frac{LK\bar{\eta}M}{\bar{B}(K-1+\delta)}\Big(\frac{K}{P} + \frac{L(2K-1)(K-1)\bar{\eta}}{6}\Big)\Big],$$

Increasing P increases the first term and hence the bound on convergence guarantee

So communications do not have to be too frequent...

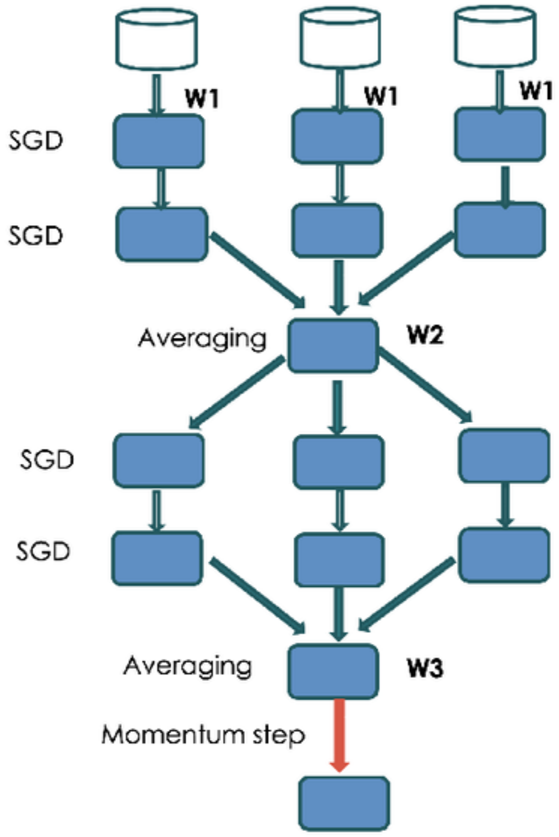
Do we just increase P for fast convergence?
Convergence challenge for large P:
Let $S = NPBK$ be constant, then

$$\frac{1}{N}\mathbb{E}\sum_{n=1}^{N}\|\nabla F(\widetilde{\mathbf{w}}_n)\|_2^2 \le \boxed{\Big[\frac{2(F(\widetilde{\mathbf{w}}_1)-F^*)PBK}{S(K-1+\delta)\bar{\eta}}} + \frac{LK\bar{\eta}M}{\bar{B}(K-1+\delta)}\Big(\frac{K}{P}+\frac{L(2K-1)(K-1)\bar{\eta}}{6}\Big)\Big],$$

Increasing P increases the first term and hence the bound on convergence guarantee

Introducing reduction momentum

```
Algorithm 2 MAVG
initialize $\widetilde{\mathbf{w}}_1$, $\mathbf{v} \leftarrow 0$
on processor $j$, $j = 1, \dots, P$, in parallel:
Learner $P_j$ set $\mathbf{w}^j_1 = \widetilde{\mathbf{w}}_1$
for $n = 1, ..., N$ do
    for $k = 1, ..., K$ do
        randomly sample a mini-batch of size $B_n$ and update:
        $$\mathbf{w}^j_{n+k} \leftarrow \mathbf{w}^j_{n+k-1} - \frac{\eta_n}{B_n} \sum_{s=1}^{B_n} \nabla F(\mathbf{w}^j_{n+k-1}; \xi^j_{k,s})$$
    end for
    $\mathbf{a} \leftarrow \frac{1}{P} \sum_{j=1}^{P} \mathbf{w}^j_{n+K}$;
    $\mathbf{d} \leftarrow \mathbf{a} - \widetilde{\mathbf{w}}_n$; $\mathbf{v} \leftarrow \mu \mathbf{v} + \mathbf{d}$;
    $\widetilde{\mathbf{w}}_{n+1} = \widetilde{\mathbf{w}}_n + \mathbf{v}$;
end for
```



MAVG convergence bound

Suppose MAVG is run with fixed step size $\eta > 0$, batch size $B > 0$ and momentum parameter $\mu \in [0, 1)$ such that the following condition holds

$$1 \geq \frac{L^2\eta^2(K+1)(K-2)}{2(1-\mu)^2} + \frac{2\eta LK}{1-\mu}$$

and

$$1 - \delta \geq L^2\eta^2/(1-\mu)^2,$$

for some constant $\delta \in (0, 1)$. Then the expected average squared gradient norms of F satisfy the following bounds for all $N \in \mathbb{N}$:

$$\begin{aligned}\sum_{n=1}^{N} \frac{1}{N}\mathbb{E}\|\nabla F(\widetilde{\mathbf{w}}_n)\|_2^2 \leq{} & \boxed{\frac{2(1-\mu)(F(w_1)-F^*)PBK}{S(K-1+\delta)\eta}} \\ & + \frac{L^2\eta^2\sigma^2(2K-1)K(K-1)}{6(K-1+\delta)B(1-\mu)^2} \\ & + \frac{2LK^2\sigma^2\eta}{PB(K-1+\delta)(1-\mu)}\left(1+\frac{\mu^2}{2(1-\mu)^2}\right) \\ & + \frac{L\eta\mu^2K^2M}{(K-1+\delta)(1-\mu)^3}.\end{aligned} \tag{1}$$

Notice how the first term is scaled by $(1-\mu)$

MAVG – optimal $\mu > 0$

Suppose MAVG is run with fixed step size $\eta > 0$, batch size $B > 0$, number of learners $P > 0$. For N meta iterations, such that

$$1 > \frac{L^2\eta^2(K+1)(K-2)}{2} + 2\eta LK$$

and

$$1 - \delta > L^2\eta^2,$$

for some constant $\delta \in (0, 1)$. When the following conditions hold,

$$\eta^2 < \frac{B(F(w_1) - F^*)}{5LN\sigma^2(5/P + 6L)} \text{ and } K \leq 5$$

or

$$1 > \frac{N\sigma^2}{2B(F(w_1) - F^*)}(\frac{1}{2LP} + \frac{1}{L}) \text{ and } K > 5$$

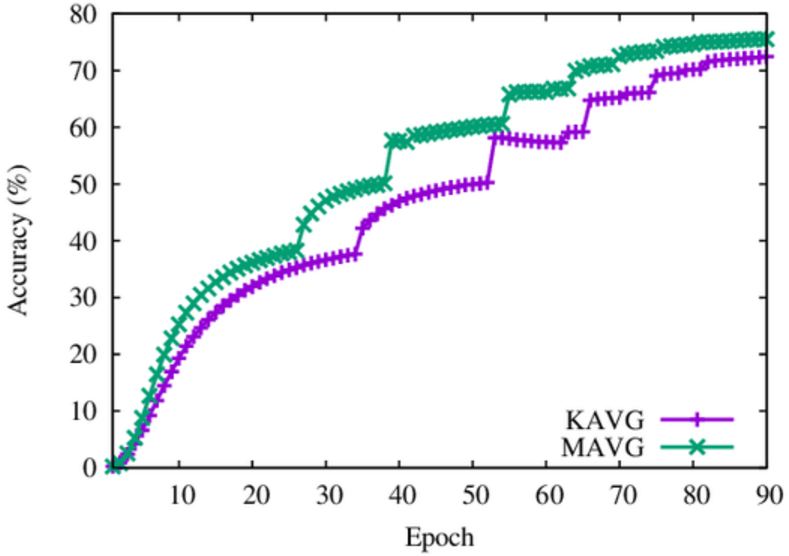
we have

$$\mu_{\text{optimal}} > 0$$

MAVG vs. KAVG

Model	KAVG	MAVG
ResNet-18	94.81	95.31
DenseNet	95.2	95.5
SENet	94.73	94.91
GoogLeNet	94.36	95.00
MoibleNet	91.77	92.16
PreActResNet-18	94.54	95.03
DPN	95.69	95.75

Test Accuracy (%), CIFAR-10, 200 epochs, P=6



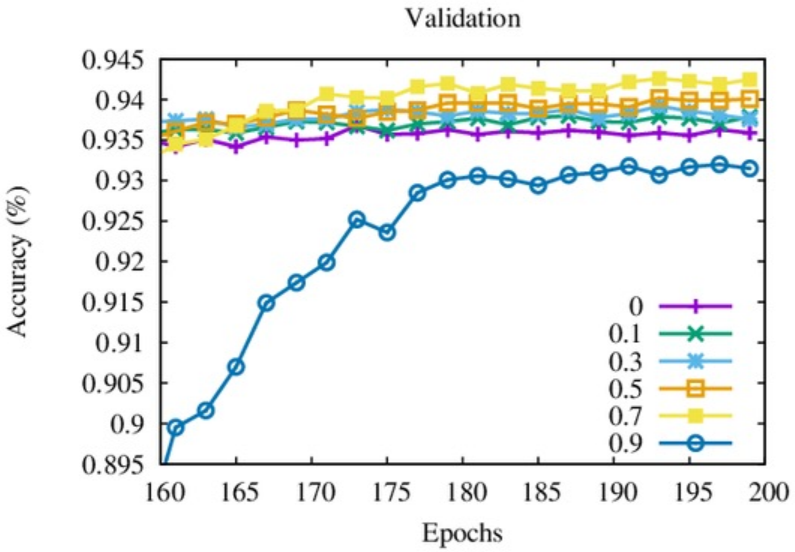
ResNet50, ImageNet-1K, P=48, μ=0.6

MAVG with regard to scaling P

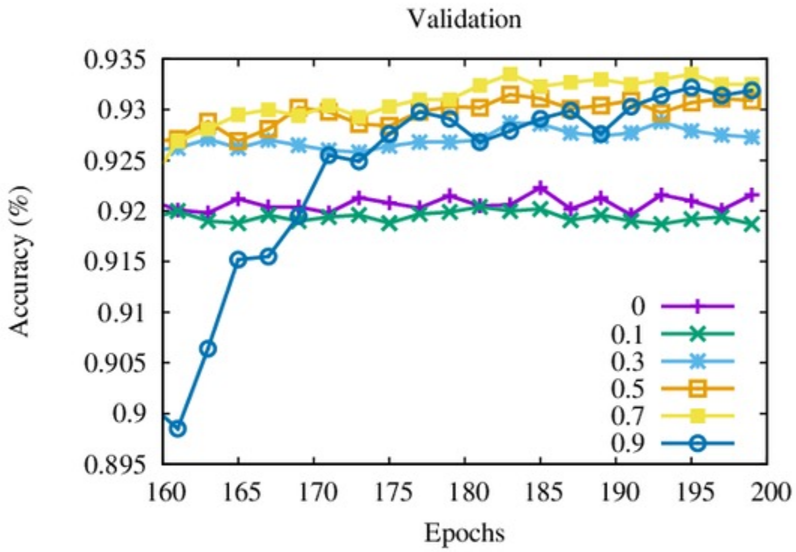
Let $S = N * P * B * K$, be a constant. Suppose MAVG is run with a fixed step size η, a fixed batch size B, and a fixed frequency K. Suppose for $P = P_0$, the optimal momentum parameter is μ_0^*. If the number of processors is increased from P_0 to λP_0, where $\lambda > 1$, the momentum parameter μ_λ^* satisfies

$$\mu_\lambda^* > \mu_0^*$$

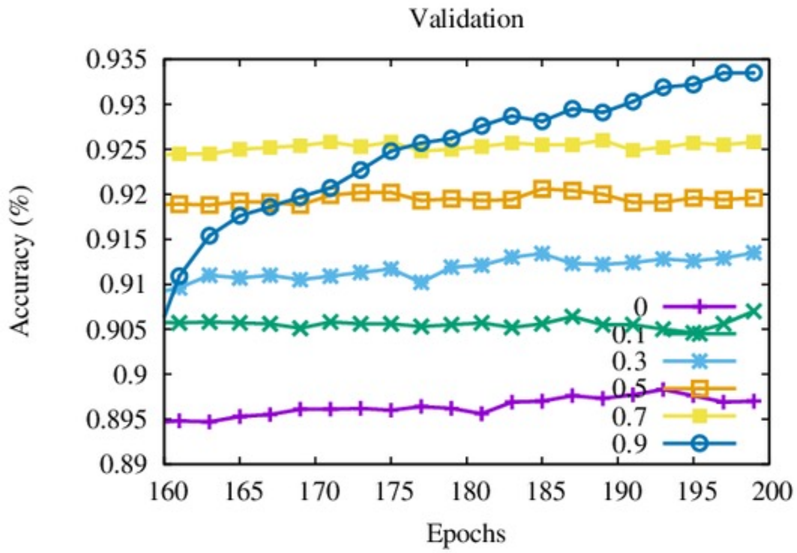
Optimal μ increases with P, CIFAR-10 and ResNet18



P=6

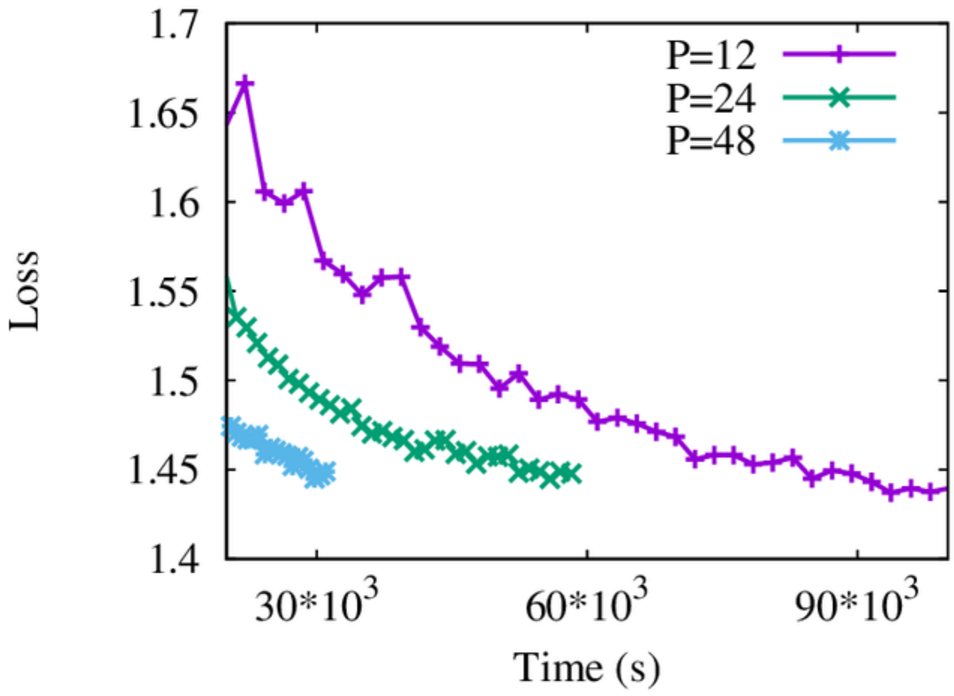


P=12



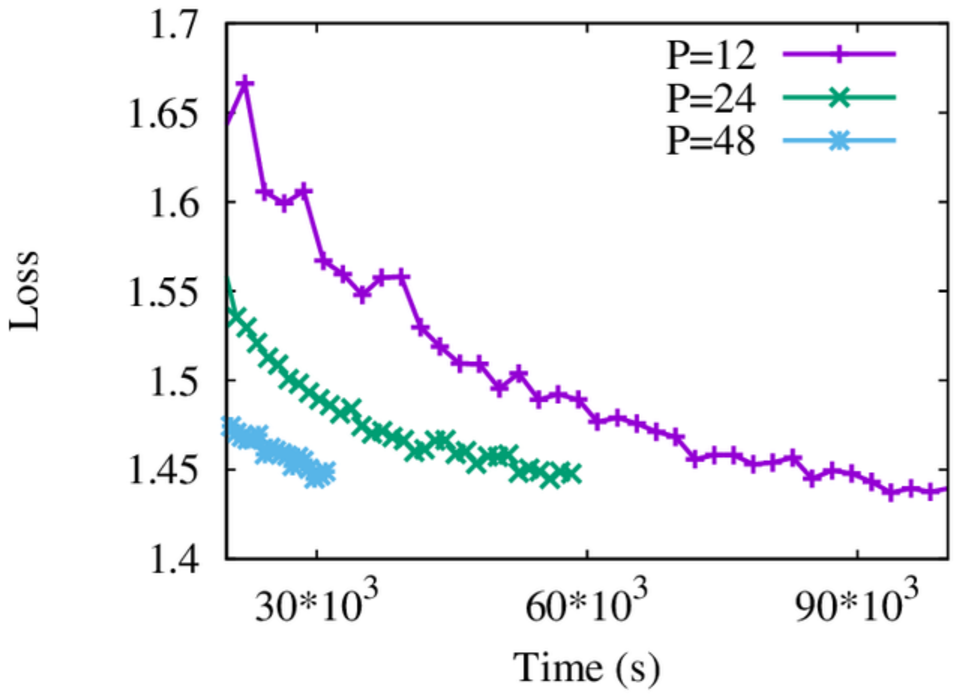
P=24

Performance on the 2000-hour speech recognition task



With 96 GPUs, MAVG trains with 3.5 hours (the previous approach takes up to a week)

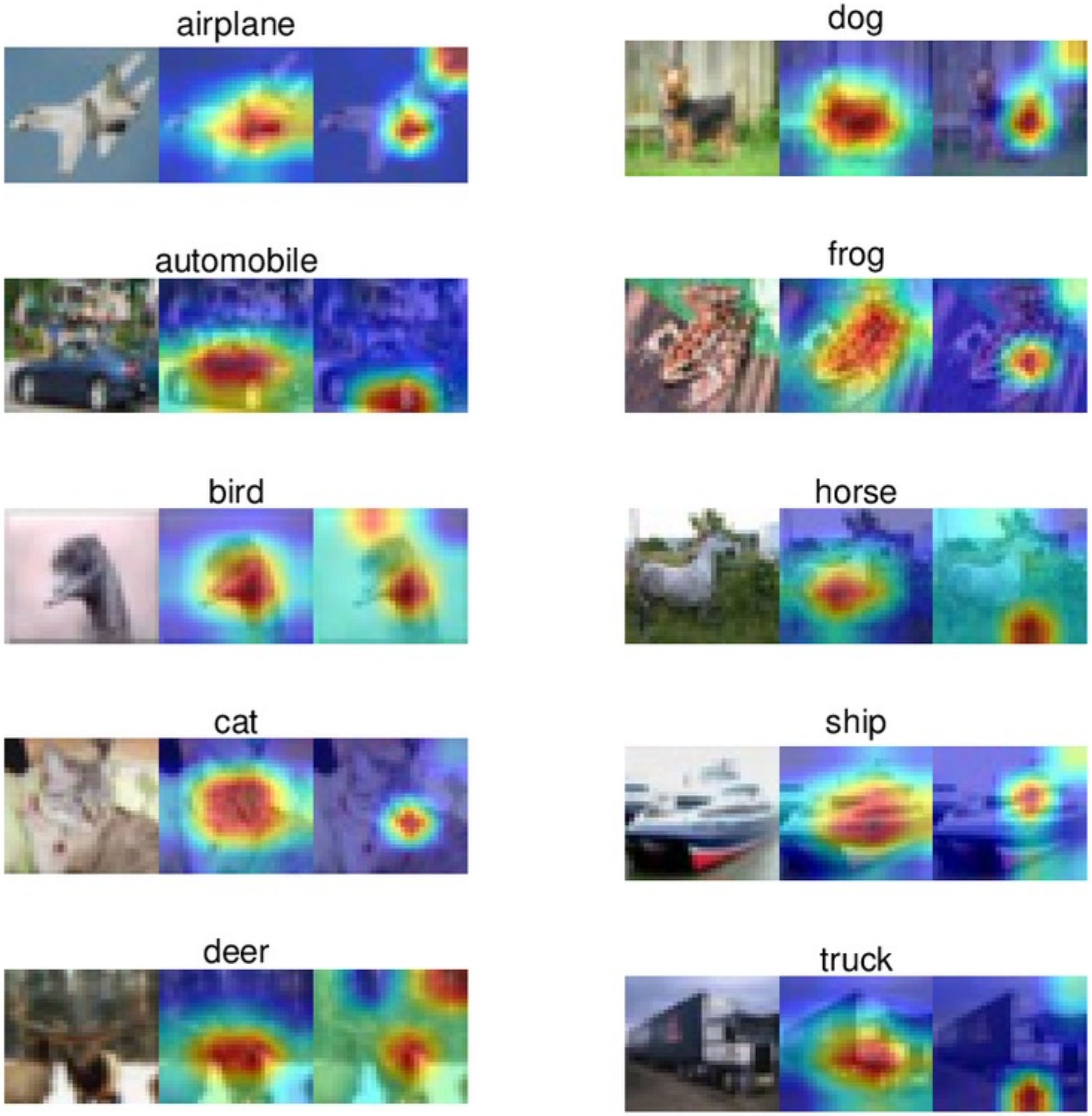
Performance on the 2000-hour speech recognition task



With 96 GPUs, MAVG trains with 3.5 hours (the previous approach takes up to a week)

Improved generalization performance

Adaptive gradient methods such as Adam tend to have poor generalization. We note that KAVG and MAVG bridge the generalization gap, as shown by the class activation mapping (CAM) that localizes important regions in the input for classification



Elastic distributed training in cloud

We proactively adjust the number of learners, and ask whether such schemes bring performance or cost advantage.

- Schedule I uses a constant number of learners P_0, $P_0 \geq 1$ – static resources
- Schedule II starts with P_0 learners and then increases to $P_1 > P_0$ learners;
- Schedule III starts with P_1 learners and then decreases to P_0 learners – Folklore choice
- Schedule IV uses a constant of P_1 learners – static resources

Evaluations

- Schedule I uses 6 GPUs and trains for 175 epochs
- Schedule IV uses 12 GPUs and trains for 350 epochs
- Schedules II and III both train for 300 epochs. In Schedule II, we start with 6 GPUs, and increase to 12 GPUs after 50 epochs. In Schedule III, we start with 12 GPUs, and decrease to 6 GPUs after 250 epochs.
- All four schedules should have similar training time. One epoch with Schedule IV takes slightly more than half the epoch time with Schedule II
- Adam optimizer, B=64, K=8

Schedules

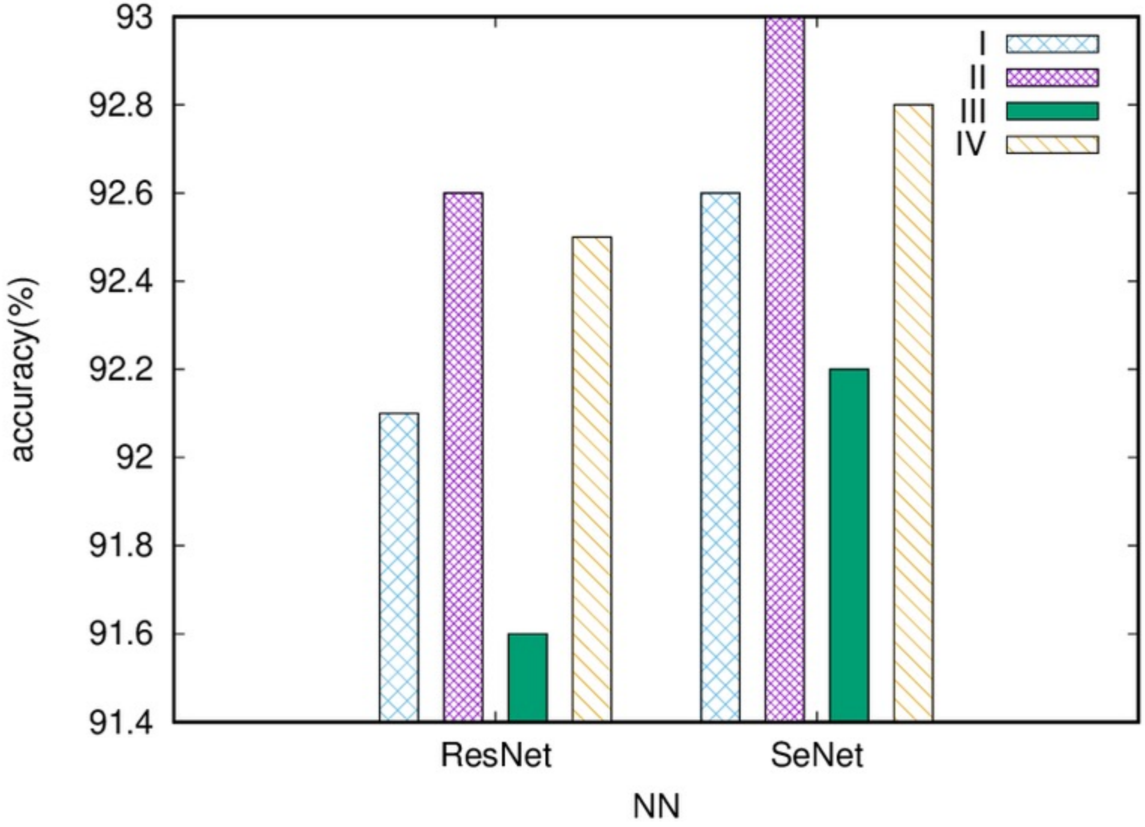


Figure: Validation accuracies for four schedules with *ResNet-18* and *SENet*

The ultimate solver

Provable fast convergence with good generalization performance for elastic resources without the need for manual tuning for current and future learning paradigms.
New challenges appear when machine learning plays an important role in simulations.

AI in Intelligent simulation for accelerated discovery

- CASTELO – drug lead optimization and immunotherapy
- MuMMi – simulating RAS proteins on cell membranes
- High-throughput screening of nanopores
- IPDYNA – Multiscale platelet dynamics for understanding of thrombosis

MuMMi – simulations of Cells and proteins for cancer cure

- Mutated RAS is found in nearly 1/3 of cancers, not yet able to target with known drugs
- Adaptive Multiscale Model, simulating RAS proteins on Cell membrane
- Machine learning directs instigation and investigation of Coarse Grain (CG) particle simulations
- Sample space more efficiently than brute force approach

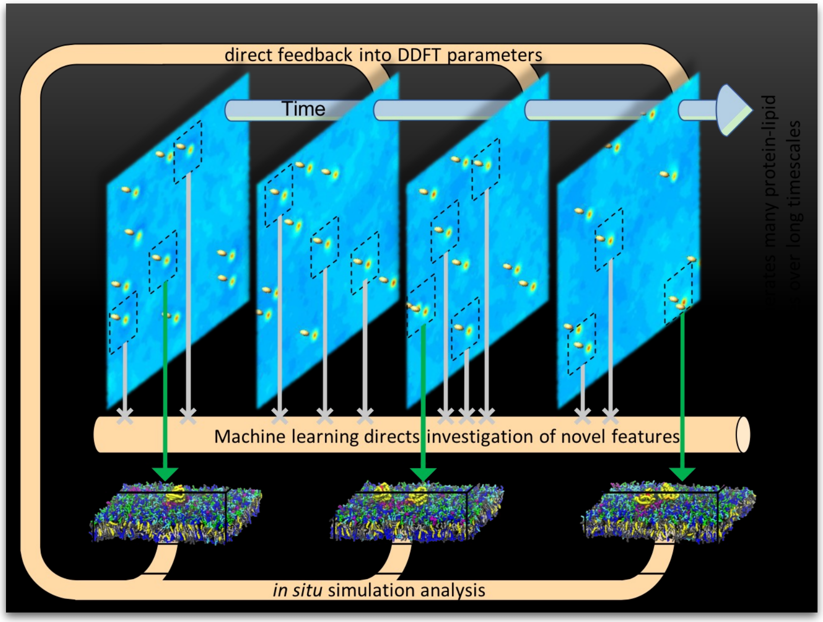
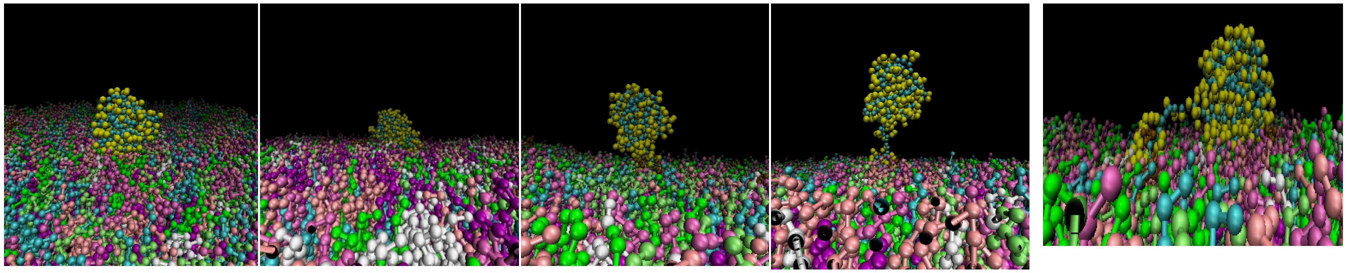
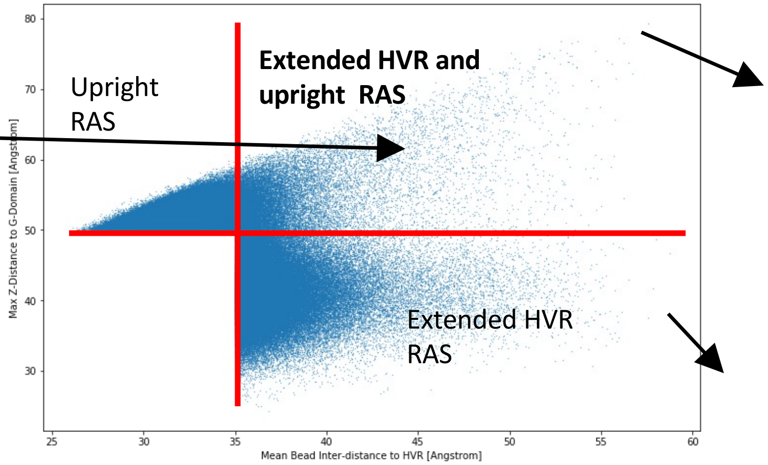

direct feedback into DDFT parameters
Time
ates many protein-lipid
over long timescales
Machine learning directs investigation of novel features
in situ simulation analysis

Image Credit: IM number LLNL-JRNL-749684

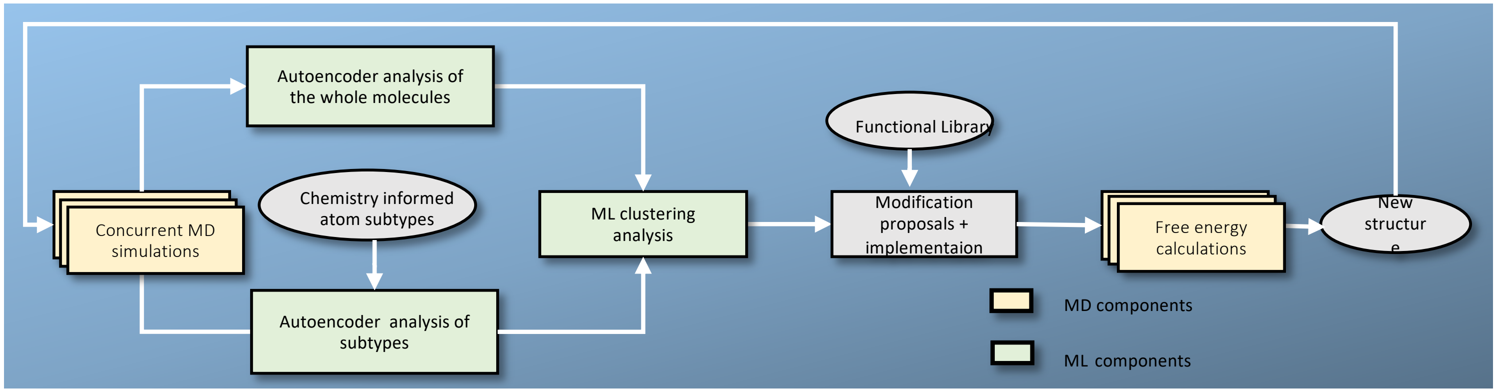
ML cataloguing events and discovering rare events



- 350TB of data generated
- Many RASProteins
- Where to instantiate the most costly detailed atomic level simulation?



Drug lead optimization



Concurrent MD simulations

Autoencoder analysis of the whole molecules

Chemistry informed atom subtypes

Autoencoder analysis of subtypes

ML clustering analysis

Functional Library

Modification proposals + implementaion

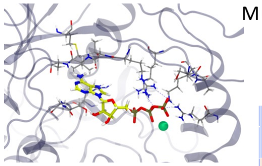
Free energy calculations

New structur e

🟨 MD components

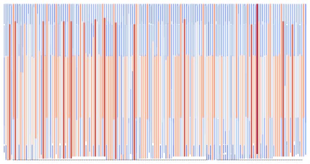
🟩 ML components

SARS-CoV-2 RdRp

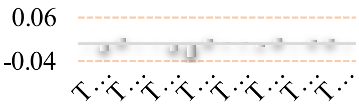


Analysis of trajectories

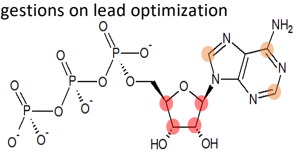
Multiple Gromacs simulations



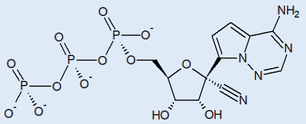
Atom subtypes ranking



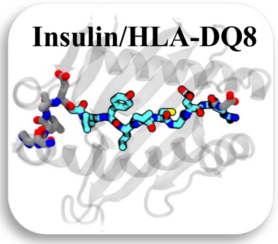
Suggestions on lead optimization

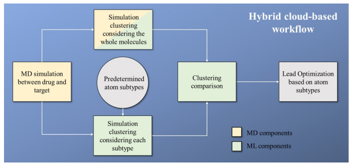


Possible outcome (~100 times stronger)

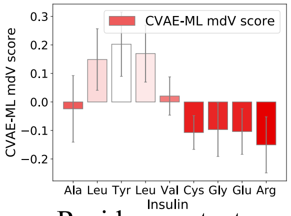


Type-1 diabetes immunotherapy

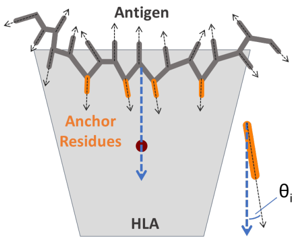



Hybrid cloud-based workflow
MD simulation between drug and target
Simulation clustering considering the whole molecules
Predetermined atom subtypes
Simulation clustering considering each subtype
Clustering comparison
Lead Optimization based on atom subtypes
MD components
ML components

CASTELO



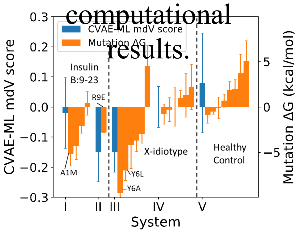
Residue contact scores



ANCHORED

Anchor residue recognition

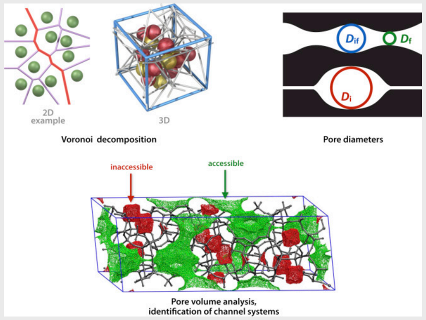
First-ever ***automated neoantigen design*** based on physics. Data agree with both experimental and computational results.

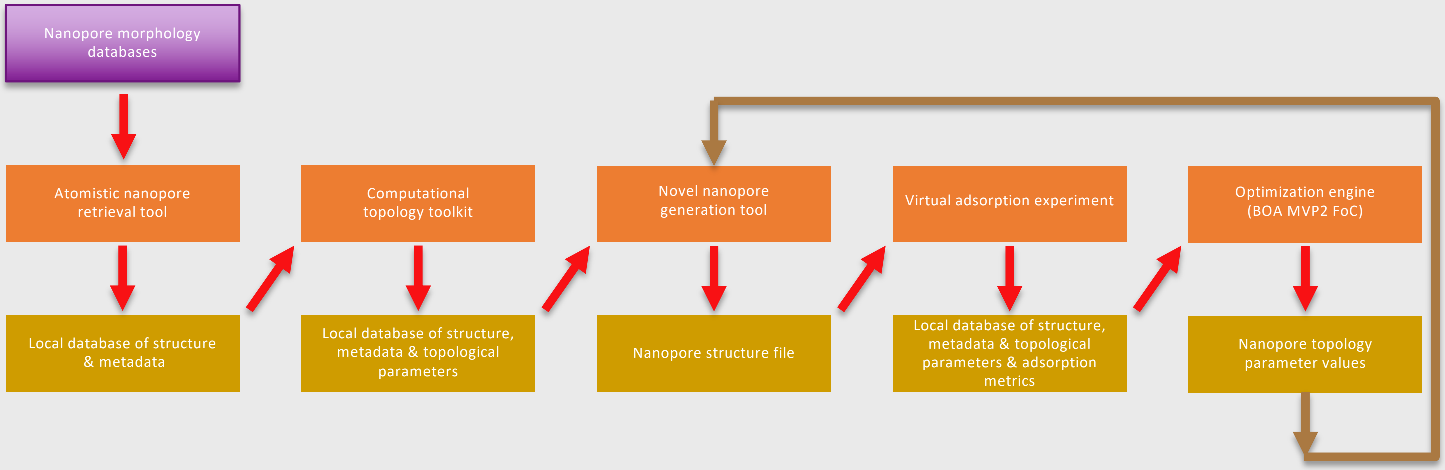


Reduces the mutation search space from 20^{10} to 20.

Vaccine designs

High-Throughput Screening of Nanopores for carbon capturing




Nanopore morphology databases

Atomistic nanopore retrieval tool

Local database of structure & metadata

Computational topology toolkit

Local database of structure, metadata & topological parameters

Novel nanopore generation tool

Nanopore structure file

Virtual adsorption experiment

Local database of structure, metadata & topological parameters & adsorption metrics

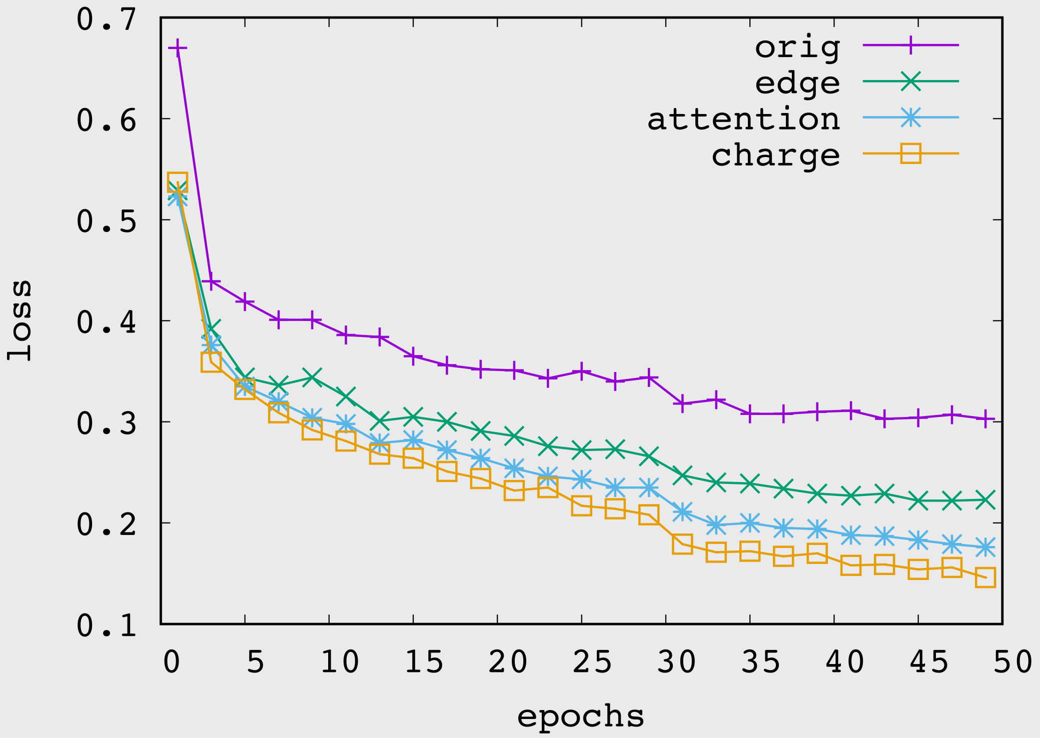
Optimization engine (BOA MVP2 FoC)

Nanopore topology parameter values

Preliminary results show that graph convolution on crystallography graph can predict adsorption of CO2

GNNs for adsorption prediction

- Orig – CGCNN
- Edge – With edge convolution
- Attention – With attention mechanism
- Charge – use atom partial charge features


loss
epochs
orig
edge
attention
charge