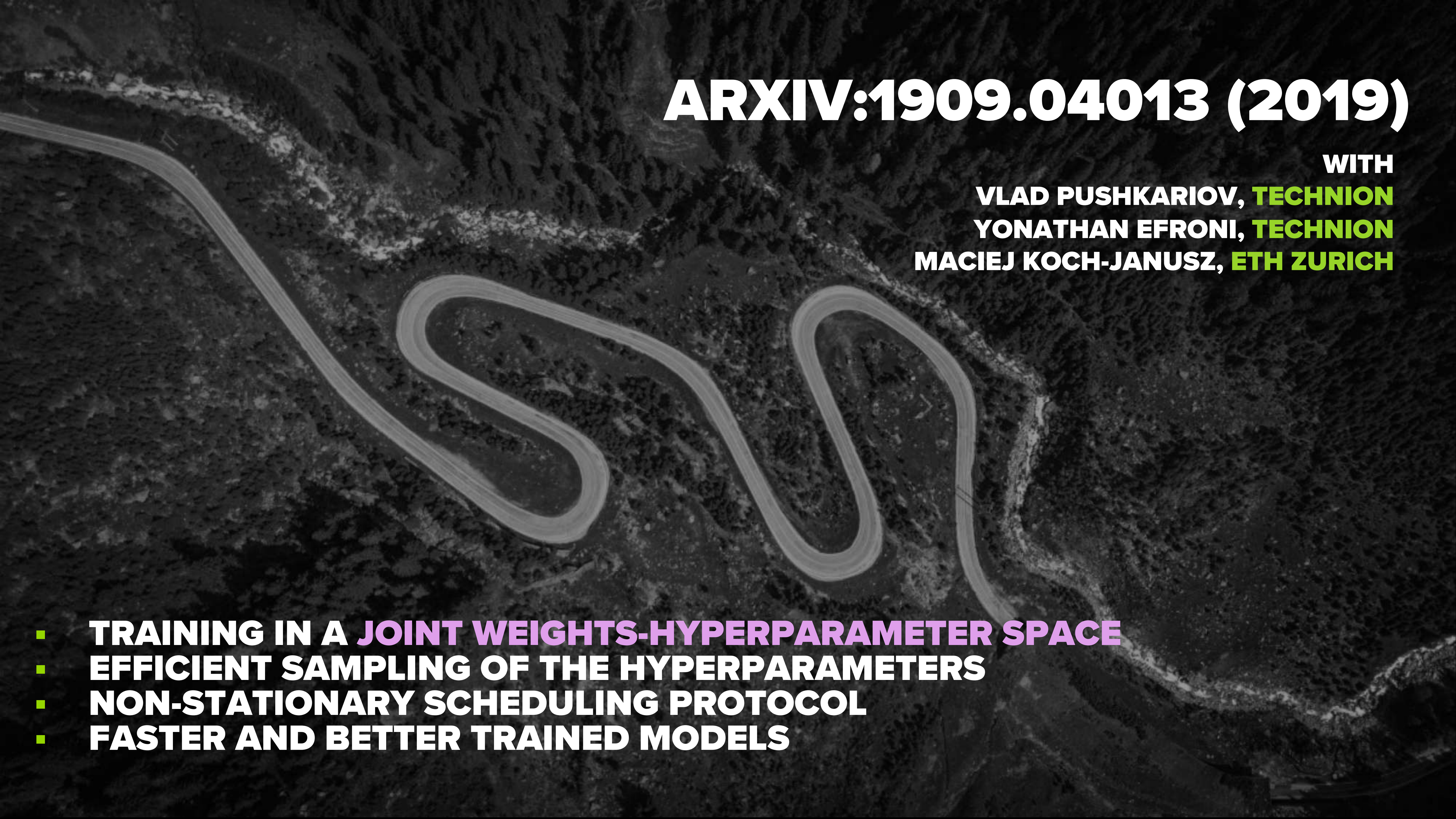
UNREASONABLE EFFECTIVENESS OF NOISE

TRAINING VIA HYPERPARAMETERS PATH OPTIMIZATION

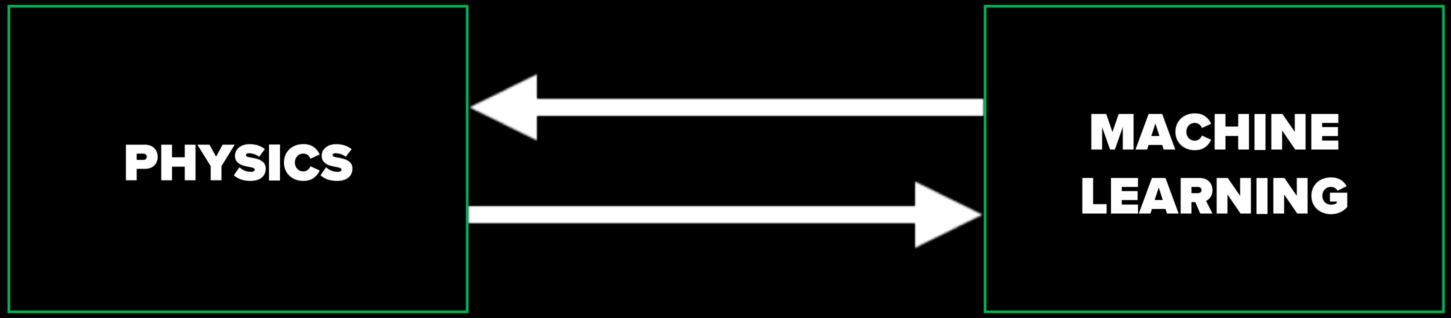
Mykola Maksymenko, R&D Director, **SoftServe**



ARXIV:1909.04013 (2019)

WITH
VLAD PUSHKARIOV, TECHNION
YONATHAN EFRONI, TECHNION
MACIEJ KOCH-JANUSZ, ETH ZURICH

- TRAINING IN A JOINT WEIGHTS-HYPERPARAMETER SPACE
- EFFICIENT SAMPLING OF THE HYPERPARAMETERS
- NON-STATIONARY SCHEDULING PROTOCOL
- FASTER AND BETTER TRAINED MODELS



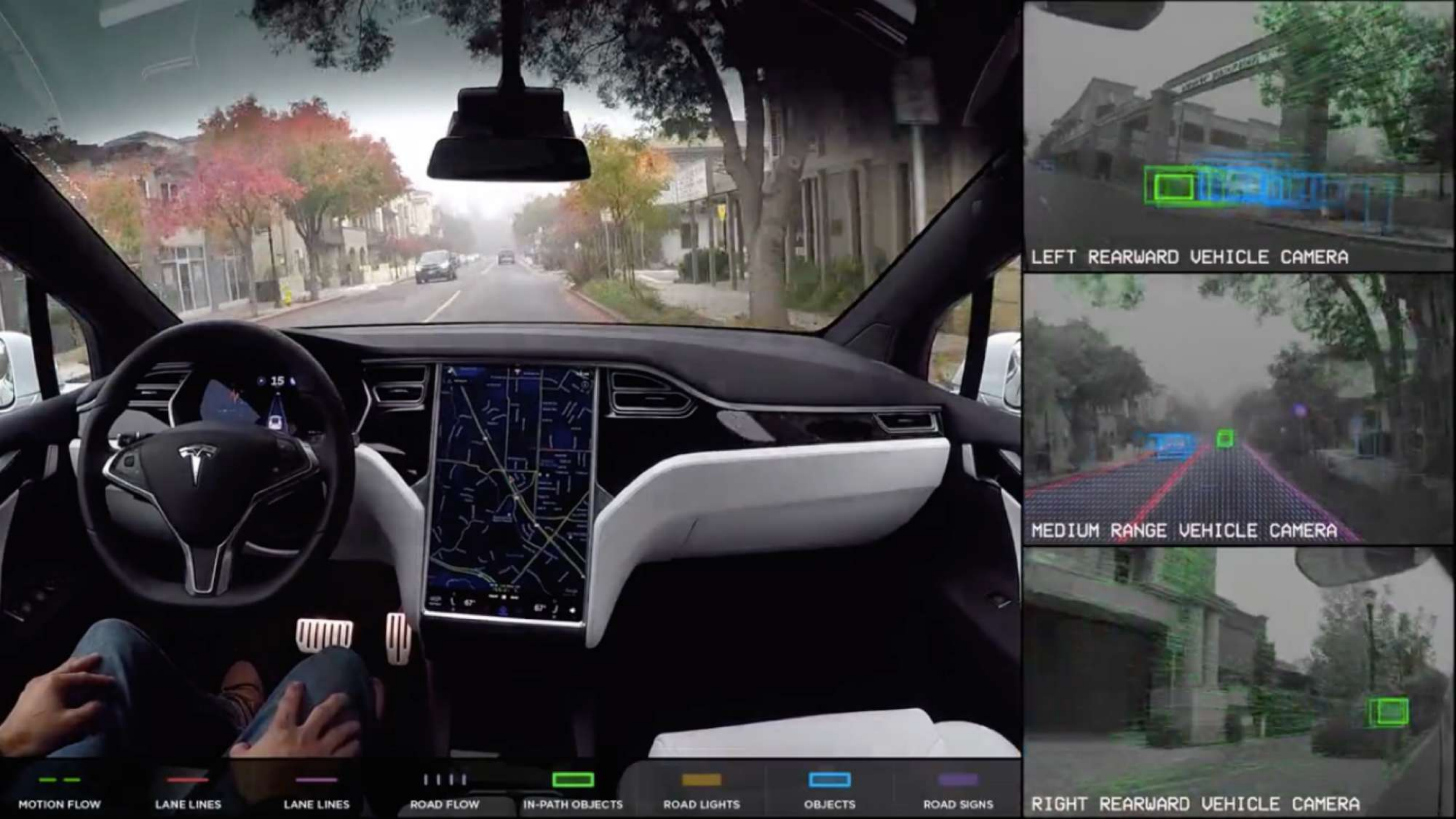
PHYSICS

MACHINE LEARNING

GOAL: OPTIMAL, SIMPLER, UNIVERSAL

PHYSICS
MACHINE LEARNING

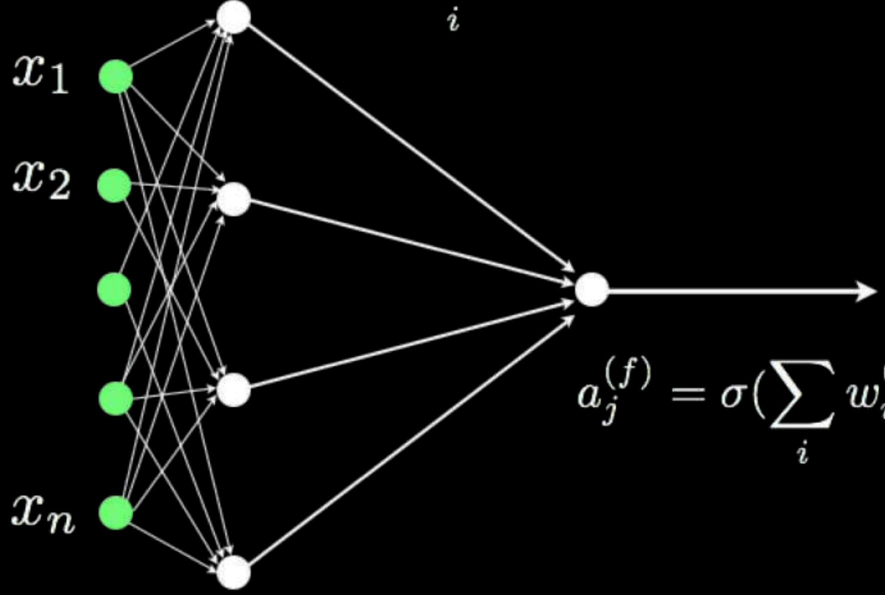
Formalism (energy, noise, etc)
Toy **models**
Concepts and **Ideas** (e.g. spin glass, tensor networks)
etc…


LEFT REARWARD VEHICLE CAMERA
MEDIUM RANGE VEHICLE CAMERA
RIGHT REARWARD VEHICLE CAMERA
MOTION FLOW
LANE LINES
LANE LINES
ROAD FLOW
IN-PATH OBJECTS
ROAD LIGHTS
OBJECTS
ROAD SIGNS

Karras et.al. arXiv:1812.04948v3 (2019)

MECHANICS OF NEURAL NETWORKS

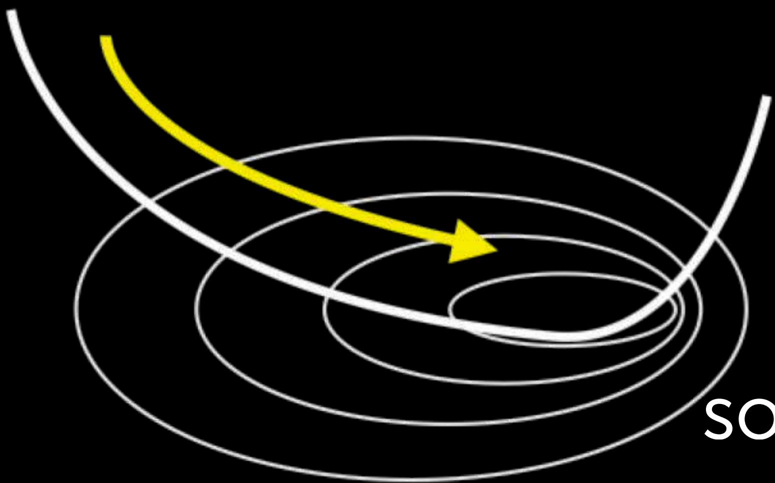
$$a_j^{(1)} = \sigma(\sum_i w_{ij}^{(1)} x_i + b_j^{(1)})$$



$$a_j^{(f)} = \sigma(\sum_i w_{ij}^{(f)} a_i^{(1)} + b_j^{(f)})$$

$$\text{Cost} = \frac{1}{2n} \sum_x ||a^f(x) - y_{\text{true}}(x)||^2$$

$$w_{ij_{t+1}} = w_{ij_t} - \gamma \nabla_{w_{ij}} Cost$$



softserve

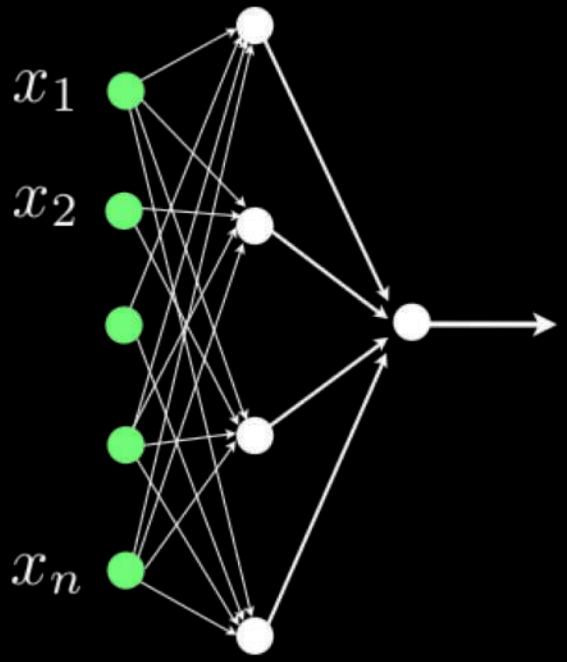
SHOULD WORK WELL IN THEORY…

Kolmogorov-Arnold representation theorem (1956-1963)

$$f(x_1, \ldots x_n) = \sum_{j=1}^{2n+1} g_j \left(\sum_{i=1}^{n} \phi_{ij}\left(x_i\right) \right)$$

Universal approximation theorem, Cybenko (1989)

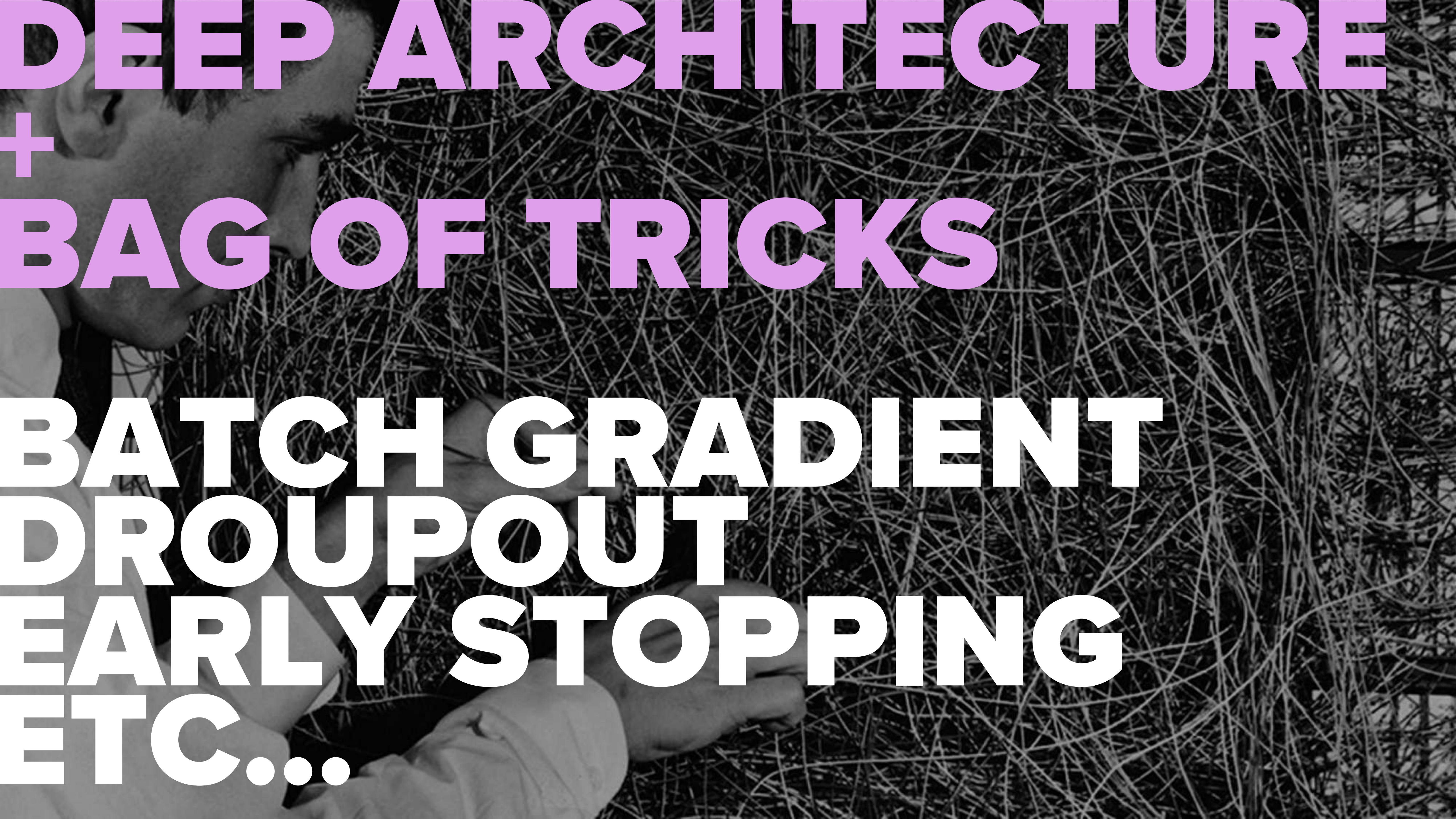
$f(\mathbf{x}) \approx$



softserve

40+ YEARS TO ADOPT IN PRACTICE

DEEP ARCHITECTURE
+
BAG OF TRICKS



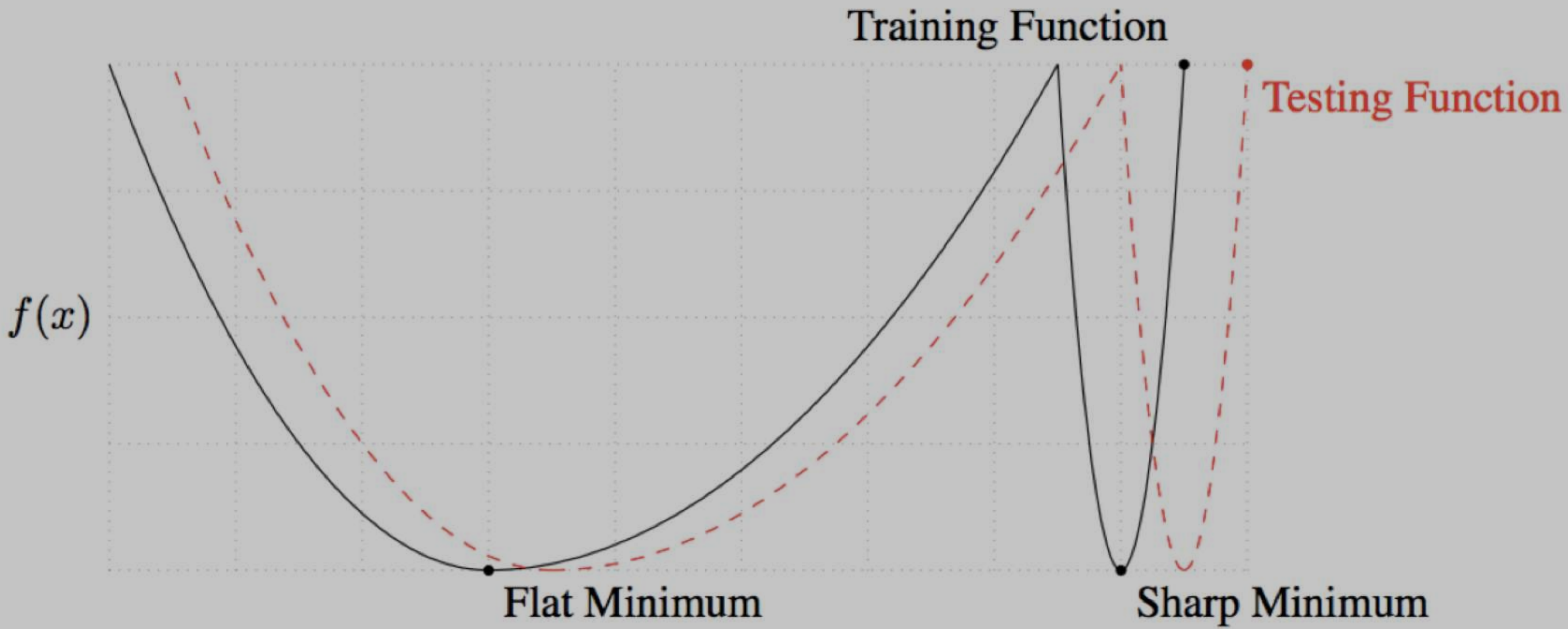
DEEP ARCHITECTURE
+
BAG OF TRICKS

BATCH GRADIENT
DROUPOUT
EARLY STOPPING
ETC…

DEEP ARCHITECTURE
+
BAG OF ~~TRICKS~~ NOISE

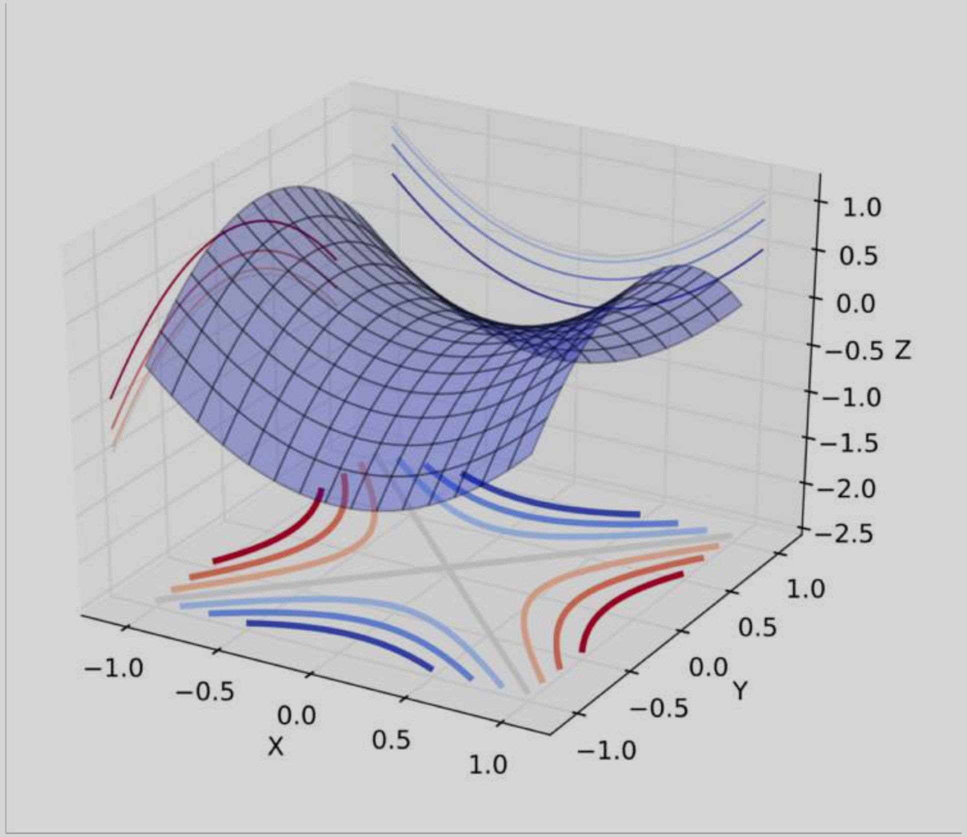
BATCH GRADIENT
DROUPOUT
EARLY STOPPING
ETC...

COMMON WISDOM: BATCH NOISE HELPS TO AVOID “BAD” MINIMA

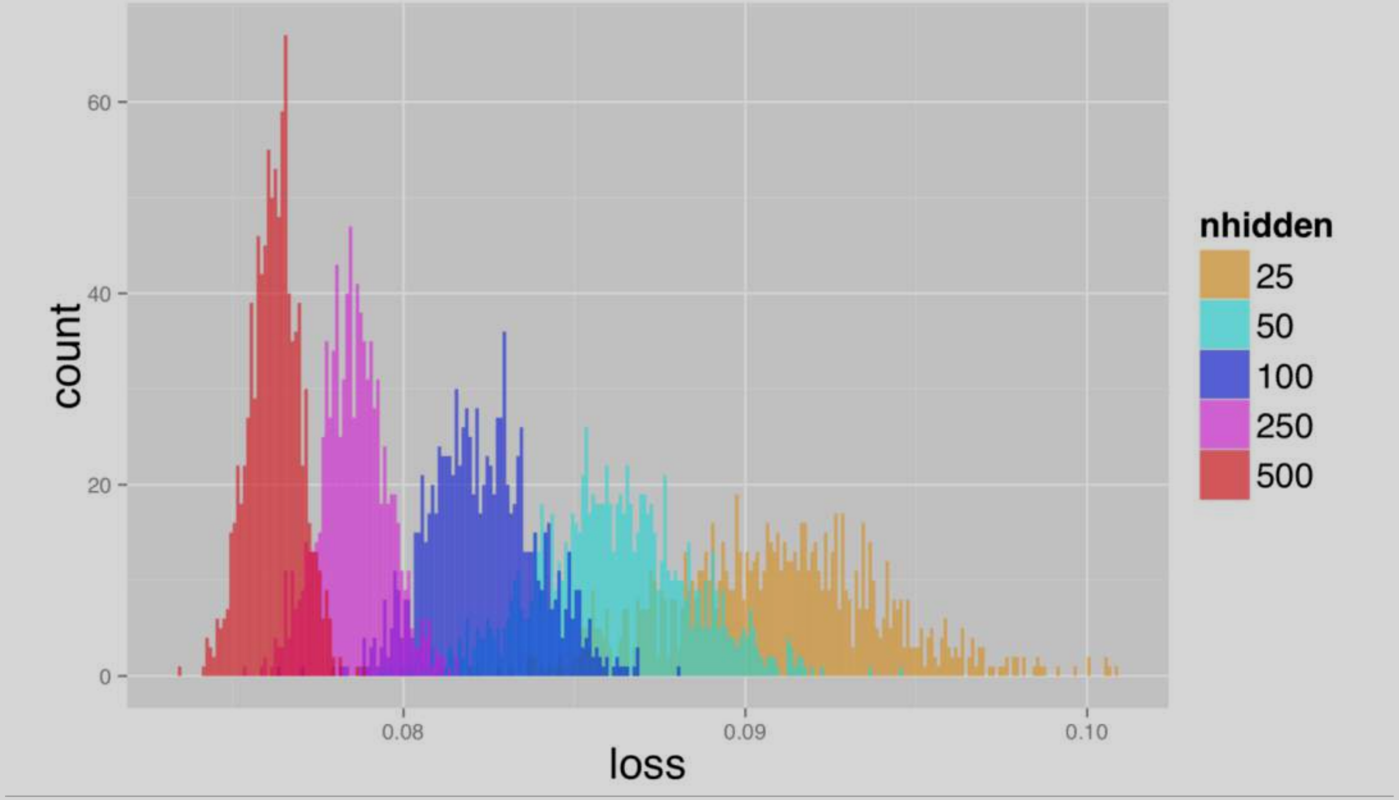


Keskar, et.al. ArXiv:1609.04836 (2017) (ICLR)

HOW LIKELY ARE THE “BAD” MINIMA?



arXiv:1406.2572v1 (2014)

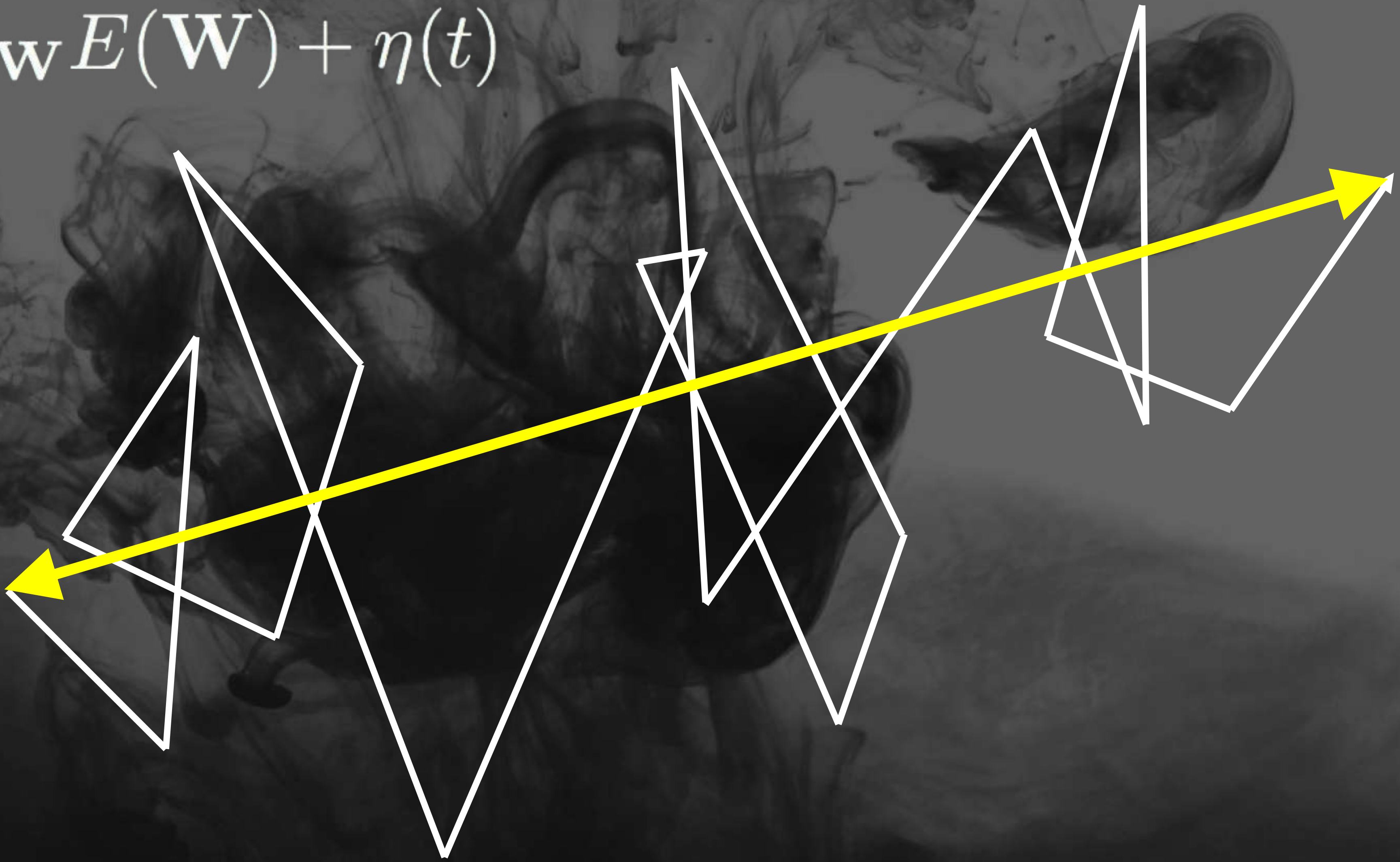


arXiv:1412.0233v3 (2015)

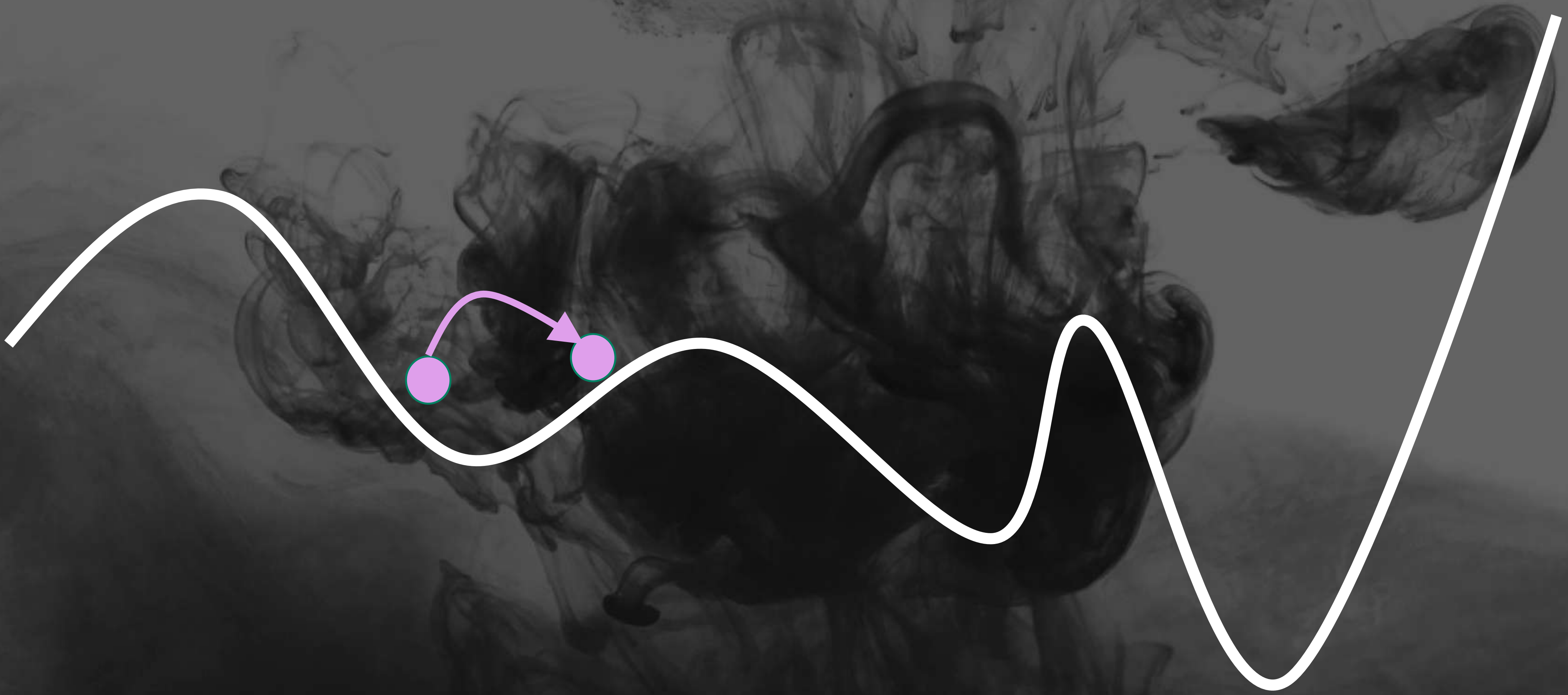
VERY UNLIKELY!

NOISE SPEEDS UP THE DIFFUSION!

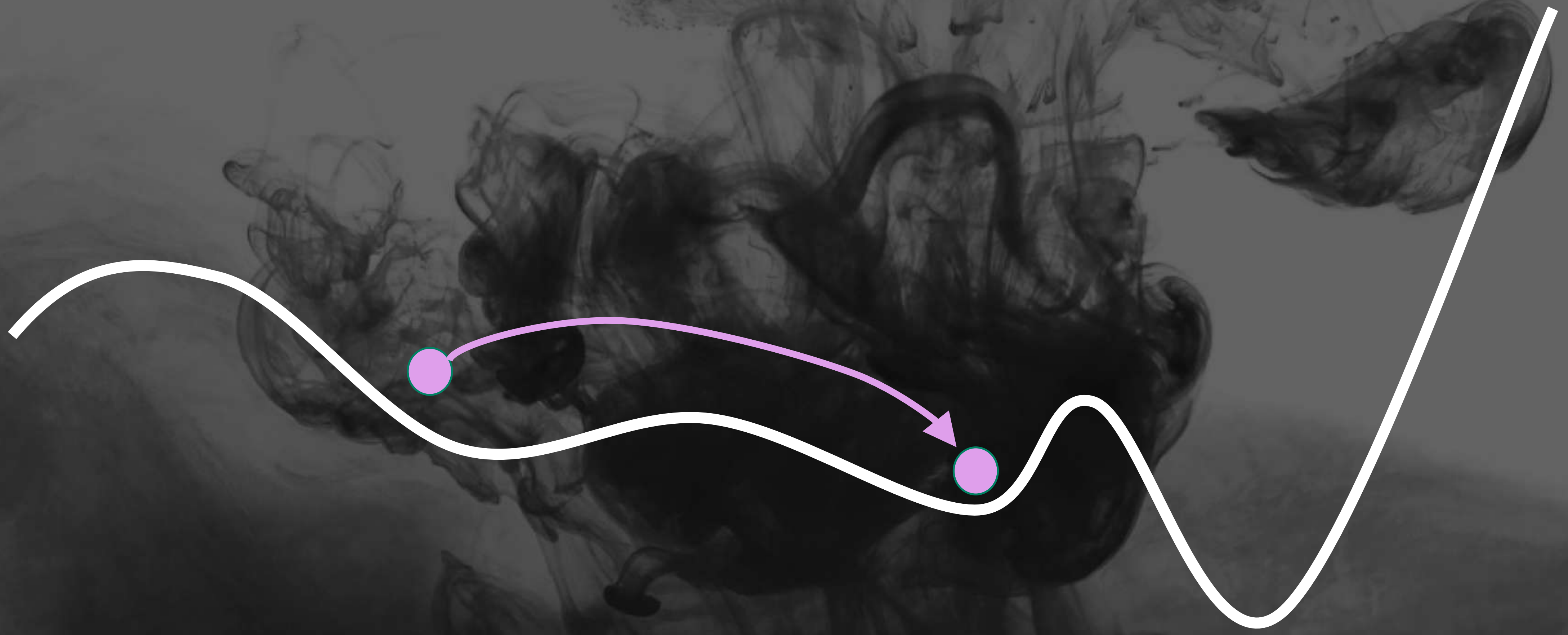
$$\frac{\partial \mathbf{W}}{\partial t} = -\nabla_{\mathbf{W}} E(\mathbf{W}) + \eta(t)$$



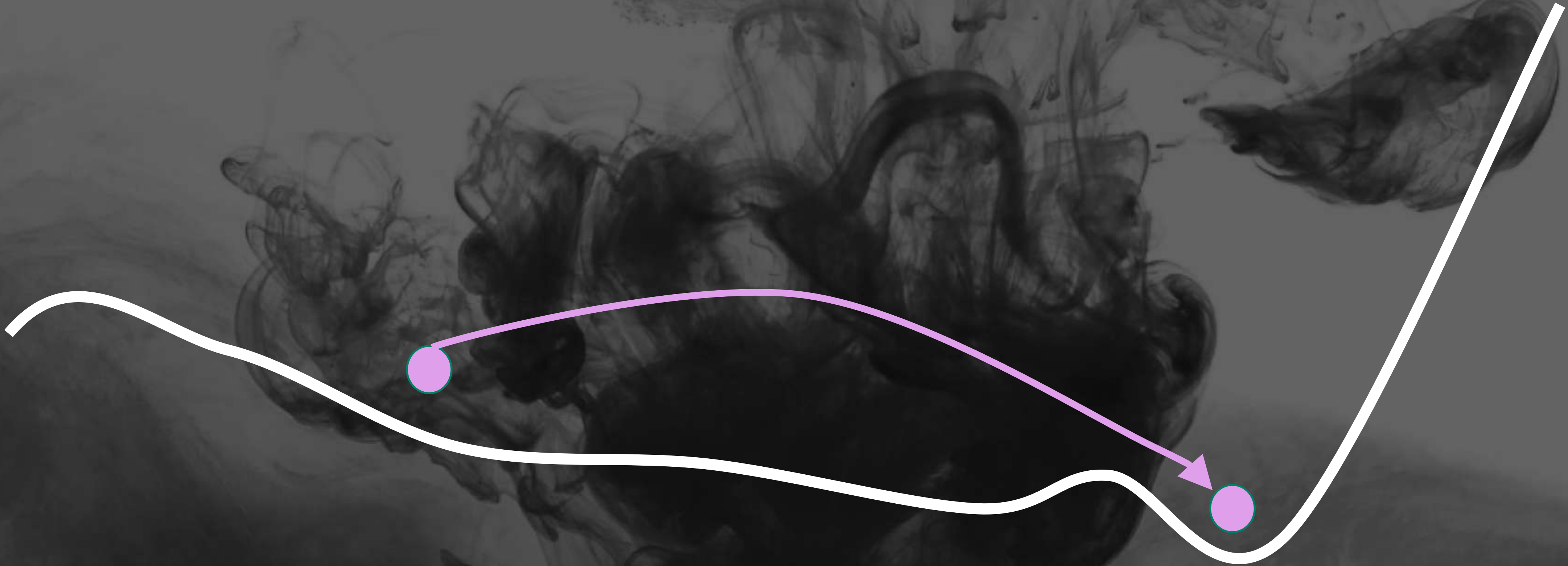
NOISE SPEEDS UP THE DIFFUSION!



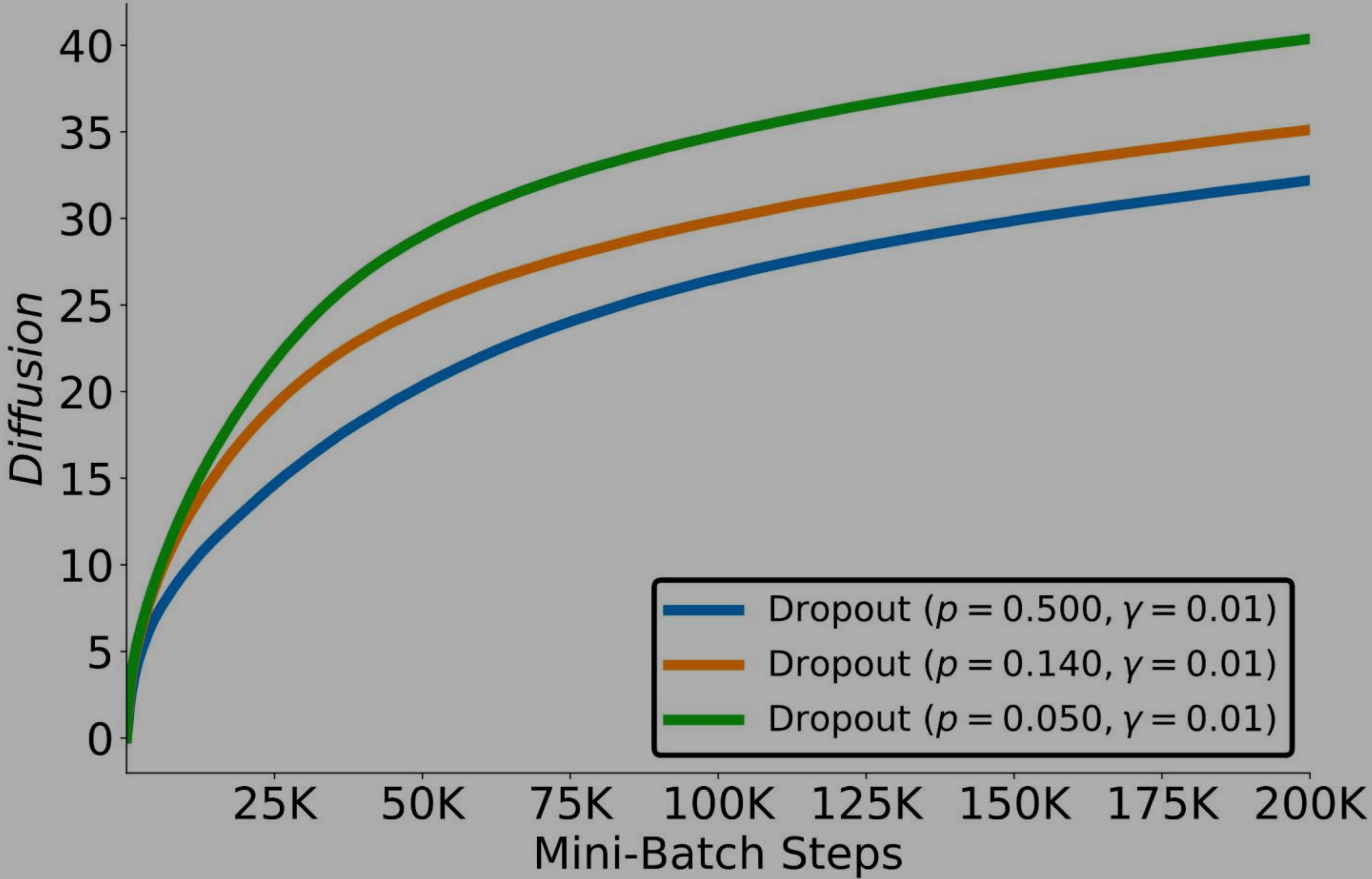
NOISE SPEEDS UP THE DIFFUSION!



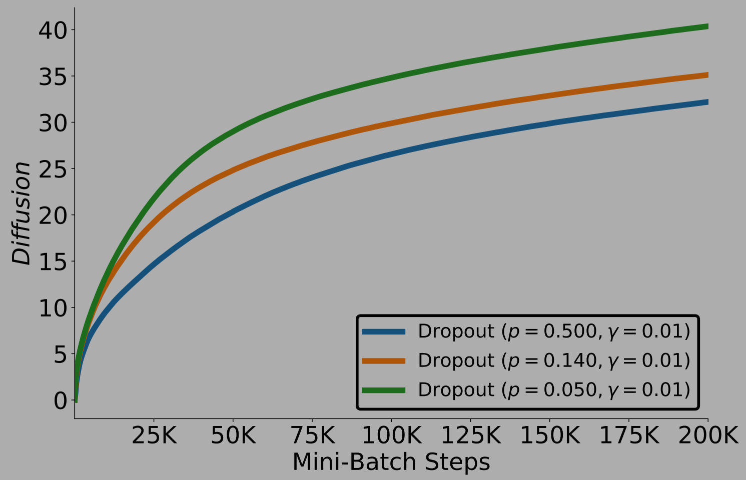
NOISE SPEEDS UP THE DIFFUSION!



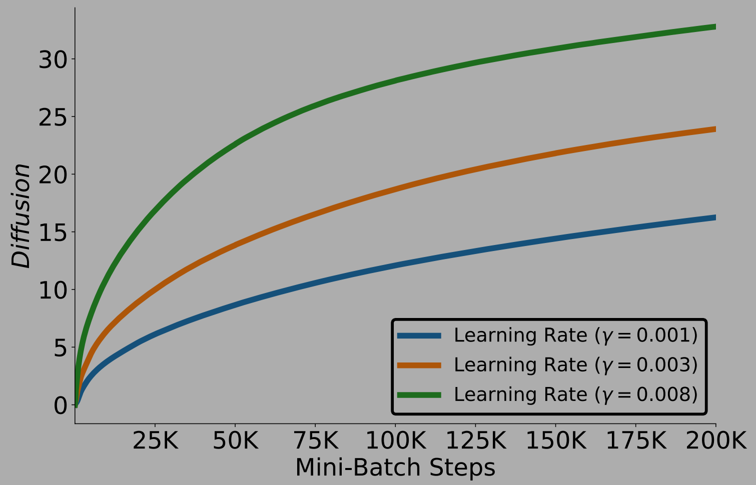
NOISE SPEEDS UP THE DIFFUSION!



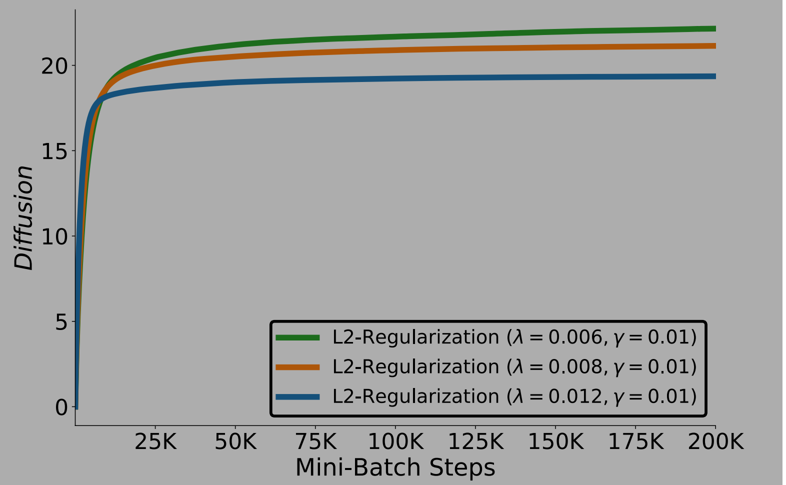
NOISE-LIKE HYPERPARAMETERS CHECK



DROPOUT ✓



LEARNING RATE ✓



L2 REGULARIZATION ✗

TRAINING VIA PATH-BASED HYPERPARAMETER OPTIMIZATION

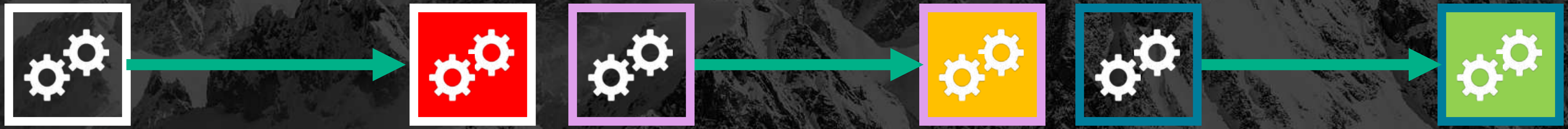
WITH

VLAD PUSHKARIOV, TECHNION
YONATHAN EFRONI, TECHNION
MACIEJ KOCH-JANUSZ, ETH ZURICH

ARXIV:1909.04013 (2019)

TYPICAL SEARCH FOR GOOD PARAMS

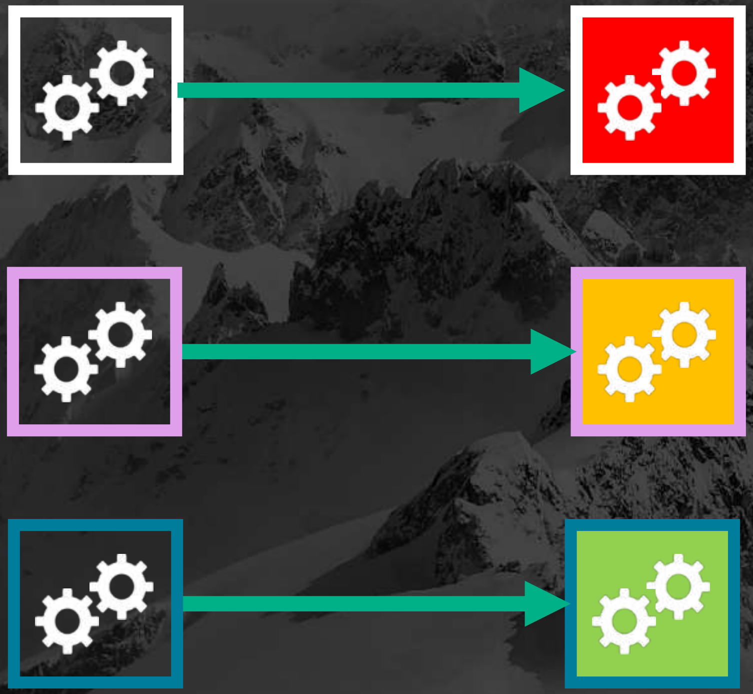
NAÏVE SEQUENTIAL SEARCH



- LONG SEARCH TIME
- LIMITS TO A SINGLE OPTIMAL SET OF PARAMETERS

TYPICAL SEARCH FOR GOOD PARAMS

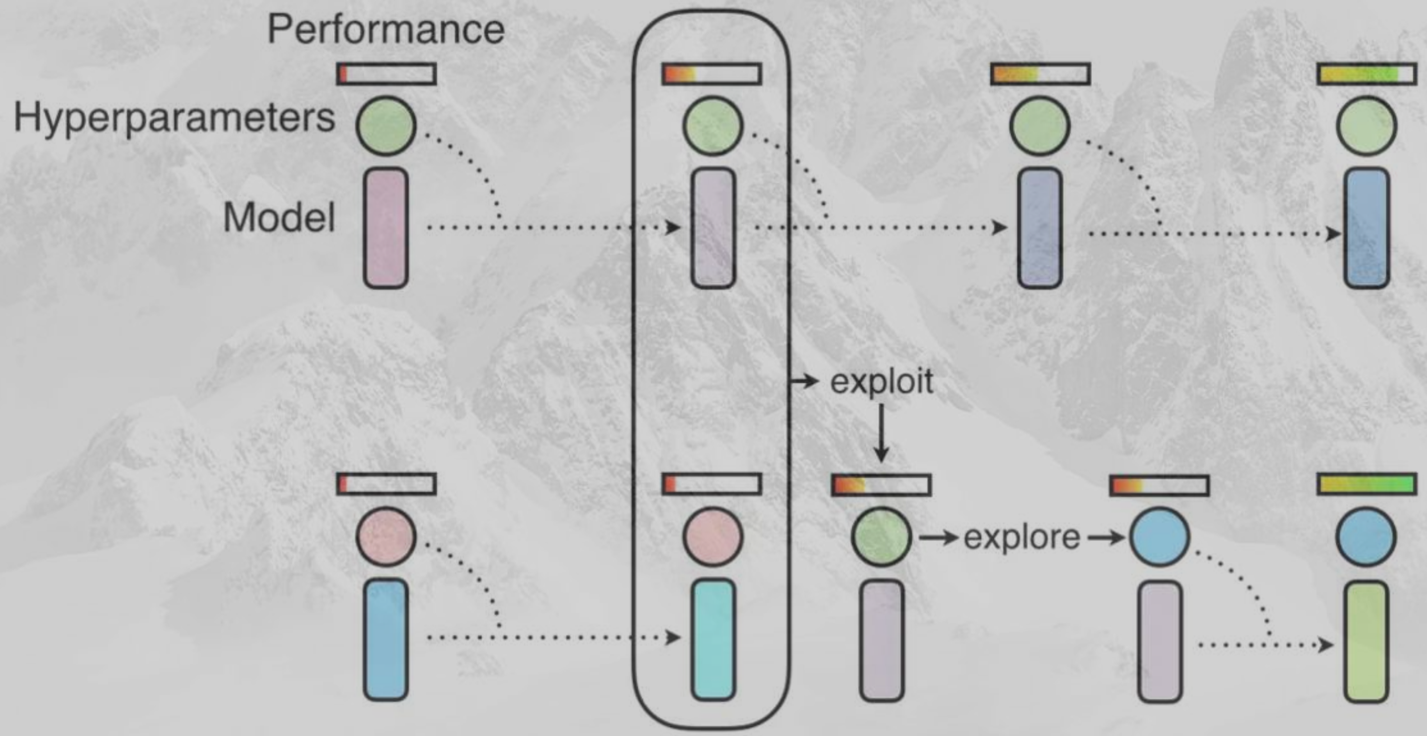
PARALLEL GRID SEARCH



- LIMITED TO A SINGLE OPTIMAL SET OF PARAMETERS

JOINT WEIGHT-HYPERPARAMS SPACE

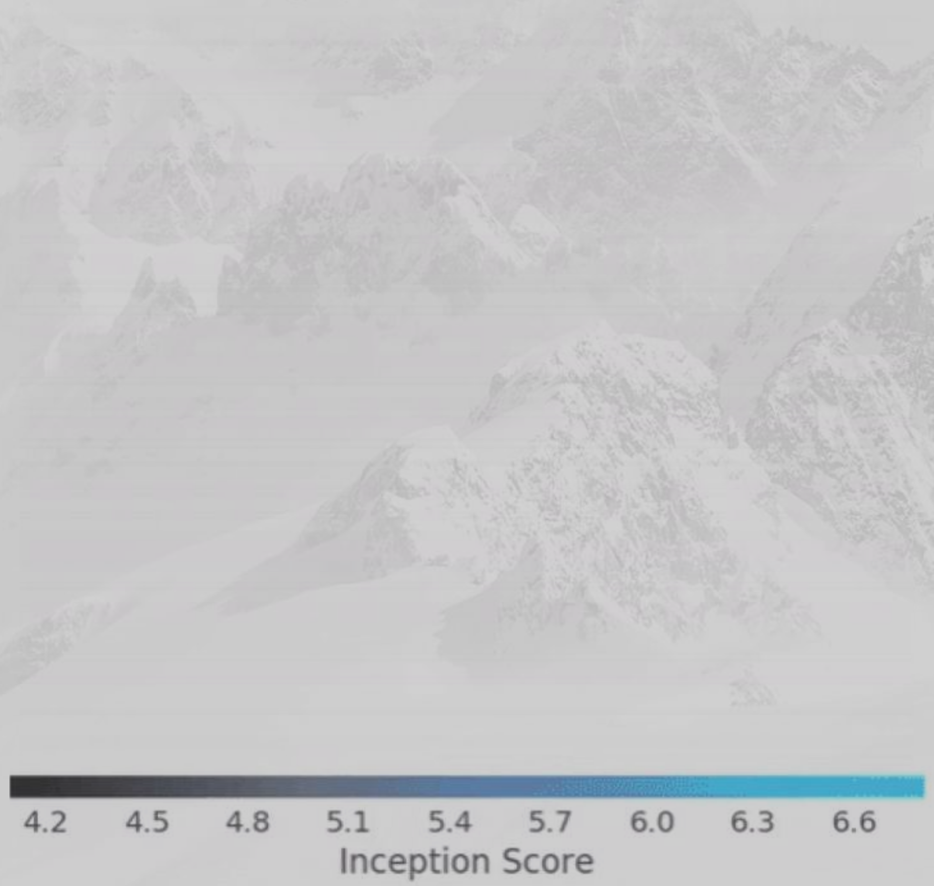
GREEDY APPROACH – POPULATION BASED TRAINING



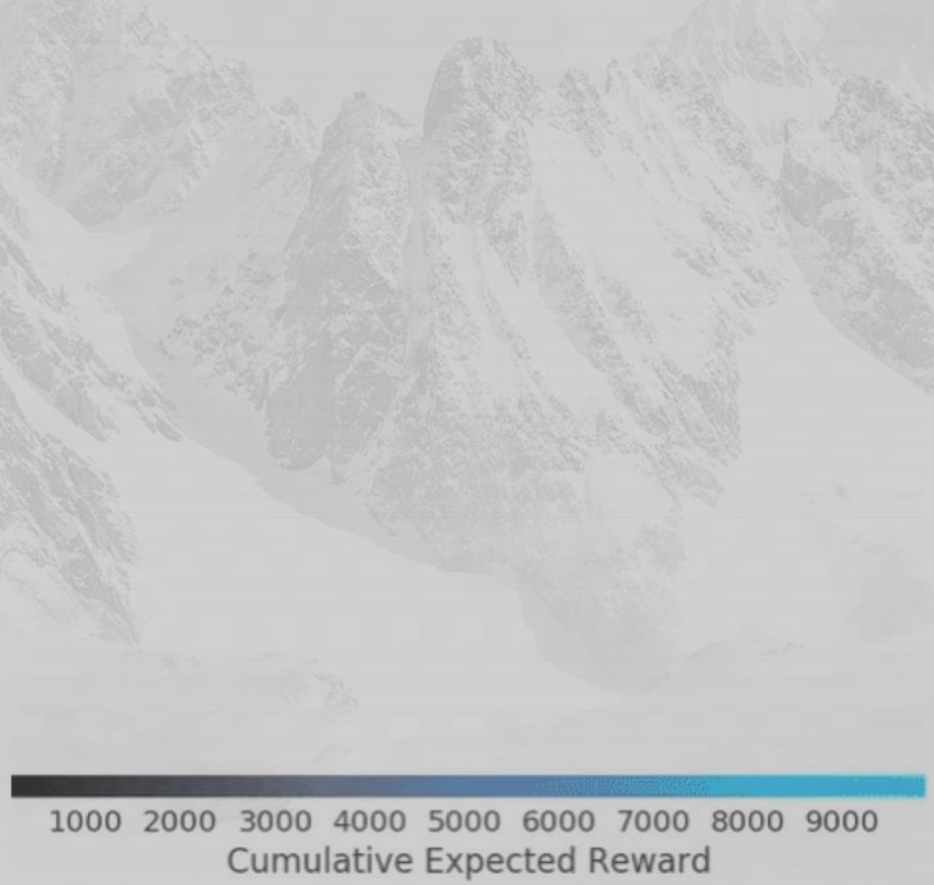
Jaderberg, et.al. arXiv:1711.09846

JOINT WEIGHT-HYPERPARAMS SPACE

GREEDY APPROACH – POPULATION BASED TRAINING



GAN population development

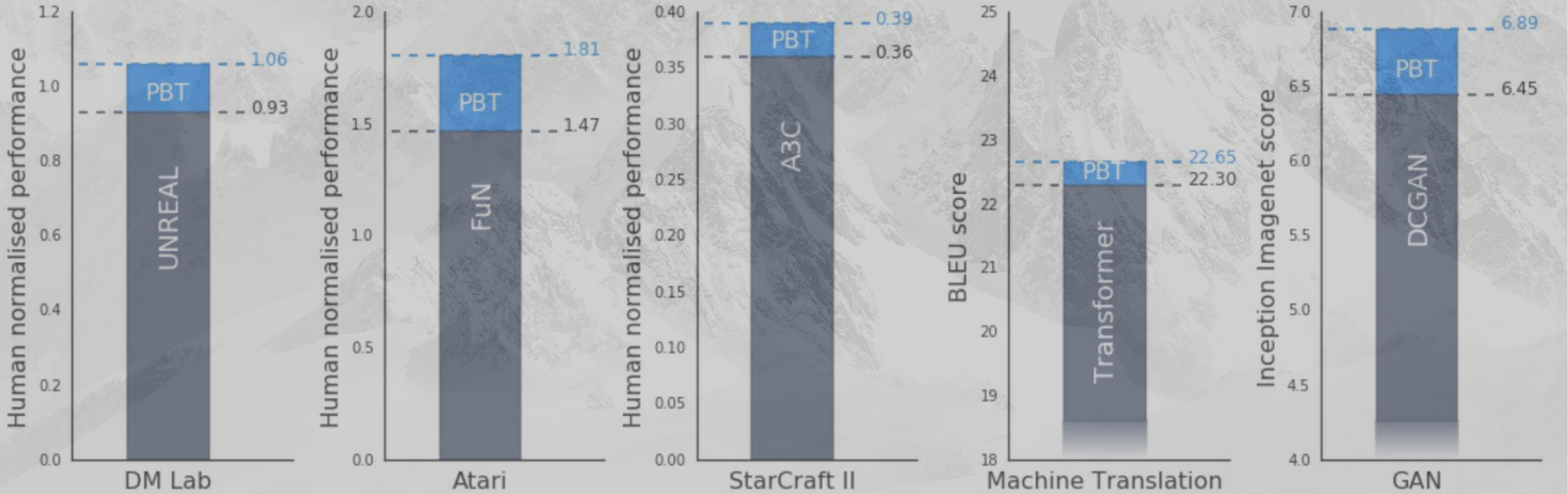


FuN population development

Jaderberg, et.al. arXiv:1711.09846

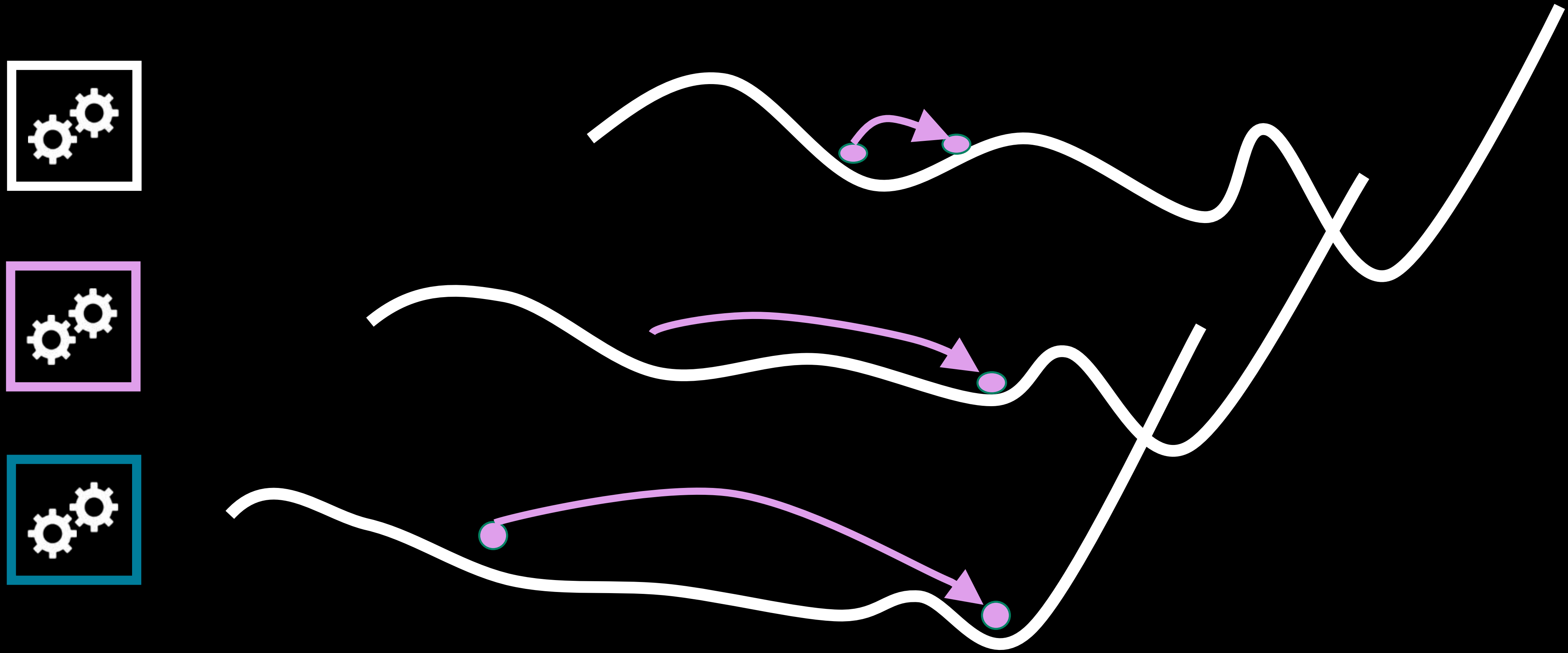
JOINT WEIGHT-HYPERPARAMS SPACE

GREEDY APPROACH – POPULATION BASED TRAINING



Jaderberg, et.al. arXiv:1711.09846

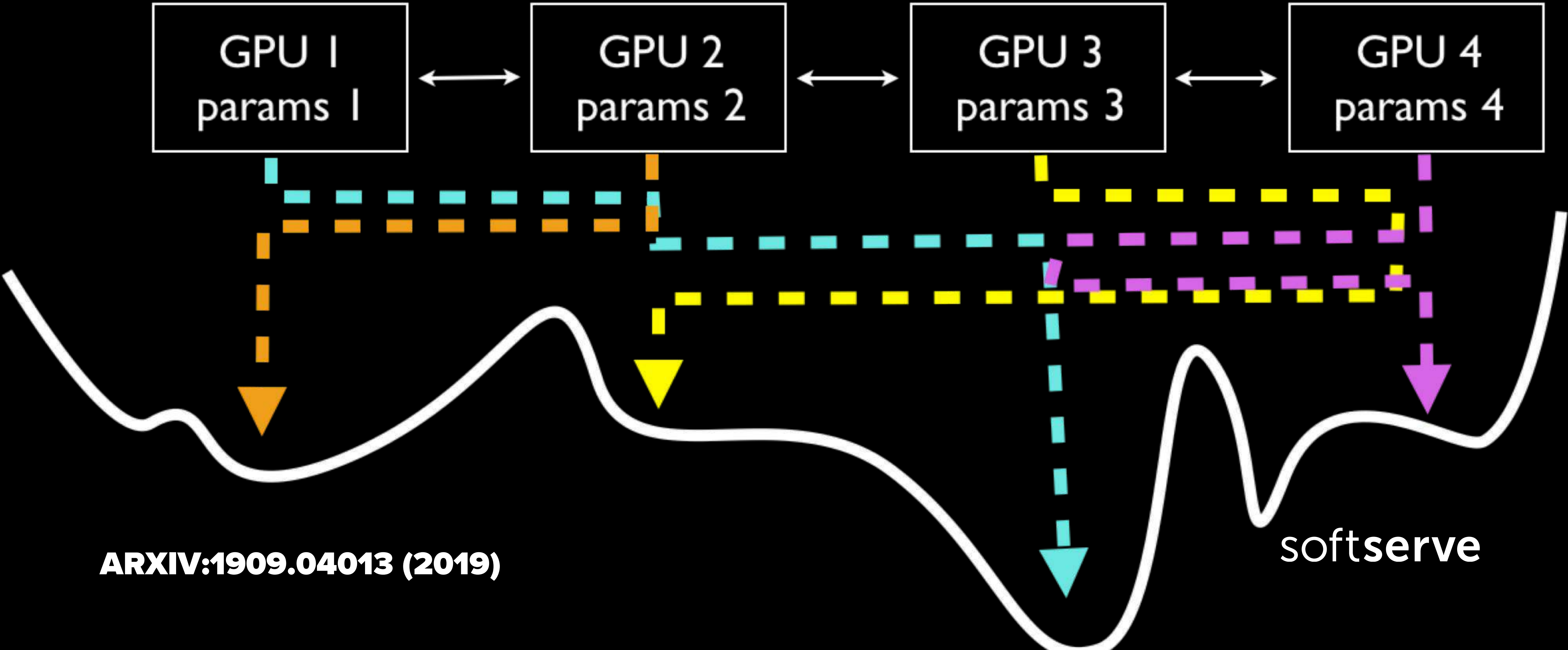
EXPLOITING NOISE-LIKE PROPERTY



ARXIV:1909.04013 (2019)

softserve

OPTIMIZATION OVER THE PATH



ARXIV:1909.04013 (2019)

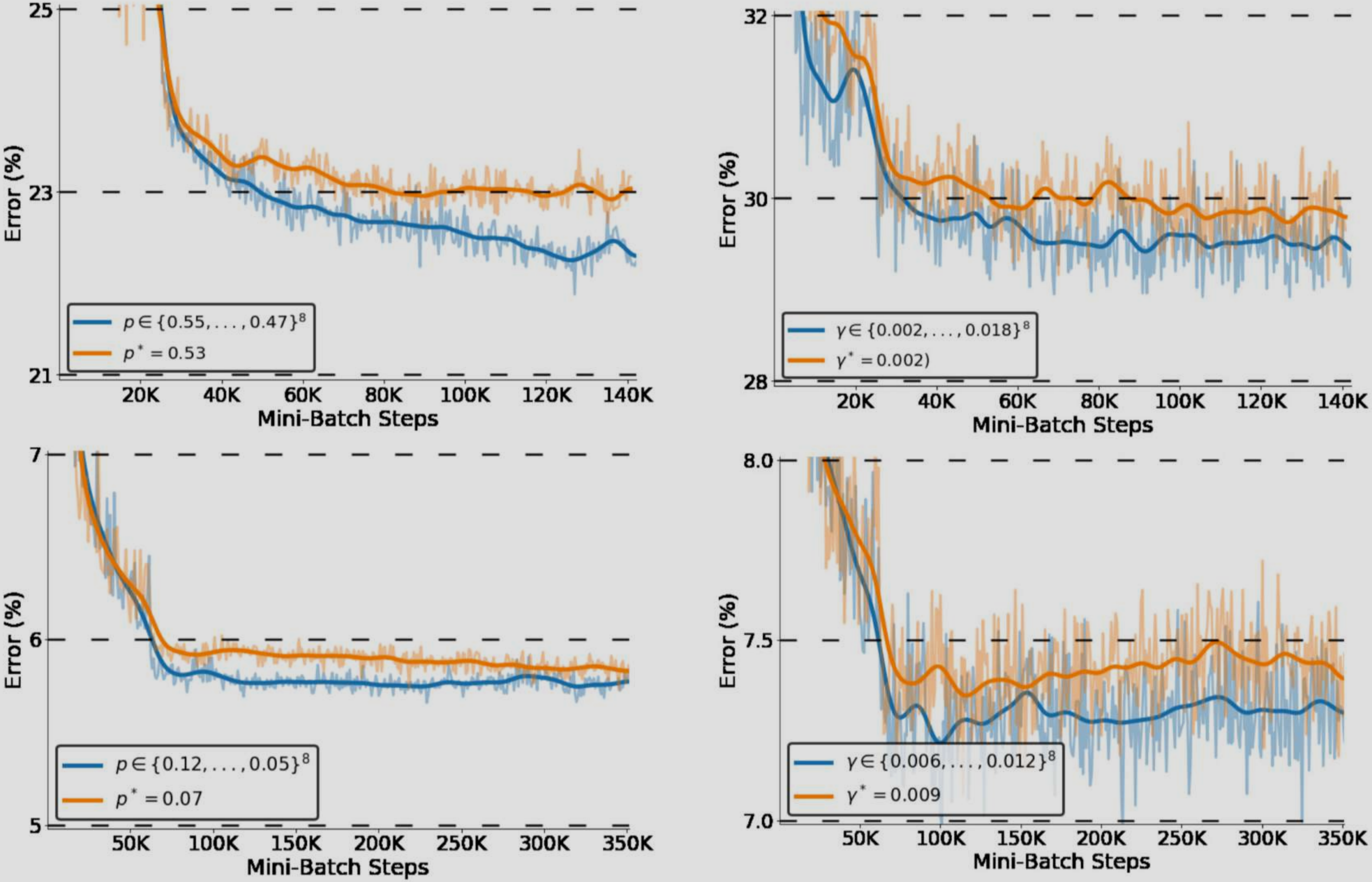
softserve

Algorithm 1 Training with replica exchange

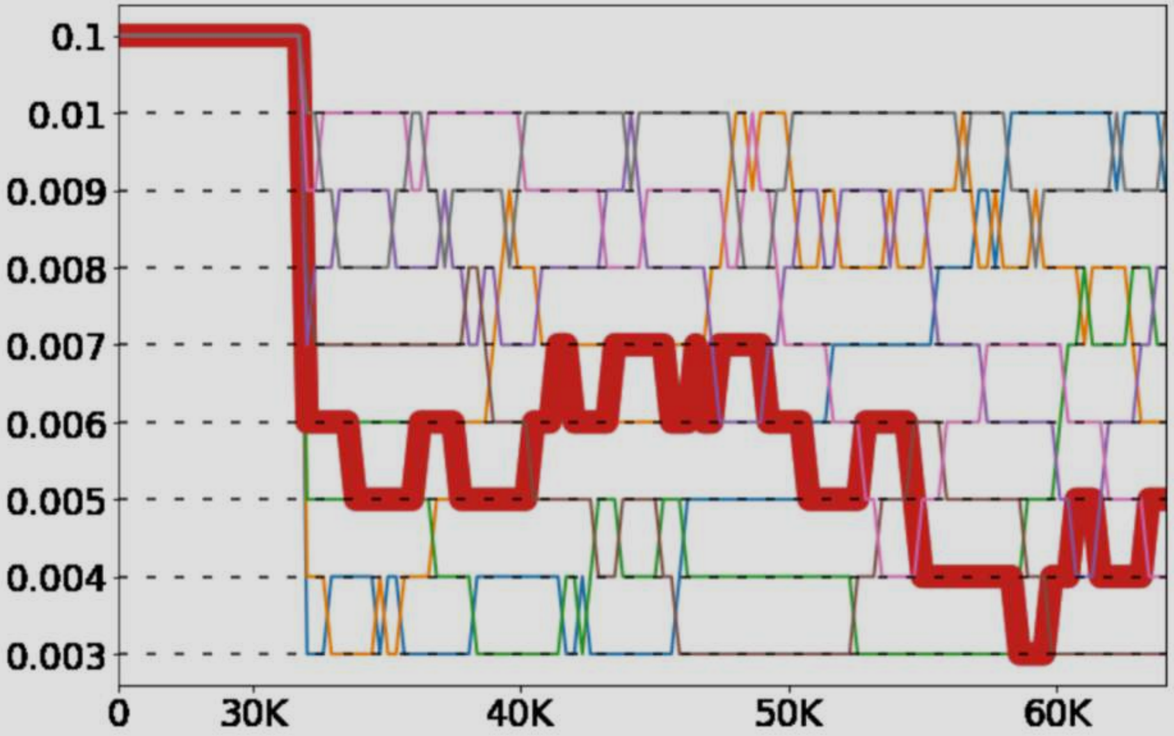
```
INPUT: Number of replicas $M$, Inverse "temperature" (hyperparameters) $\beta = (\beta_1, \beta_2, \ldots, \beta_M)$
       Number of steps for initialization $\Delta N_i$
       Number of SGD steps between exchanges $\Delta N_e$
       Exchange normalization parameter $C$
       Number of steps $T$
OUTPUT: Weight configurations $\mathbf{W} = (\mathbf{W_1}, \mathbf{W_2}, \ldots, \mathbf{W_M})$ of the replicas,
Initialization: $\forall k \in M$, initialize weights $\mathbf{W}_k$ for each replica and set $t = 0$.
$\forall k \in M$, perform SGD for $\Delta N_i$ steps. Update $t \leftarrow t + 1$ at each step.
Repeat:
    $\forall k \in M$, perform SGD for $\Delta N_e$ steps to update $\mathbf{W}_k$. Set $t \leftarrow t + 1$ at each step.
    Let $\mathcal{L}_t = (\mathcal{L}(\mathbf{W}_1^t), \mathcal{L}(\mathbf{W}_2^t), ..., \mathcal{L}(\mathbf{W}_k^t))$ be validation losses at time $t$.
    Randomly select a pair $(m, n)$ of replicas with adjacent temperatures.
    if $\Delta = C(\beta_m - \beta_n)[\mathcal{L}(\mathbf{W}_m) - \mathcal{L}(\mathbf{W}_n)] \leq 0$ then
        swap $\beta_m$ and $\beta_n$
    else
        swap $\beta_m$ and $\beta_n$ with probability $\exp(-\Delta)$.
    Update $\alpha$, the acceptance ratio. Finish if $t > T$.
```

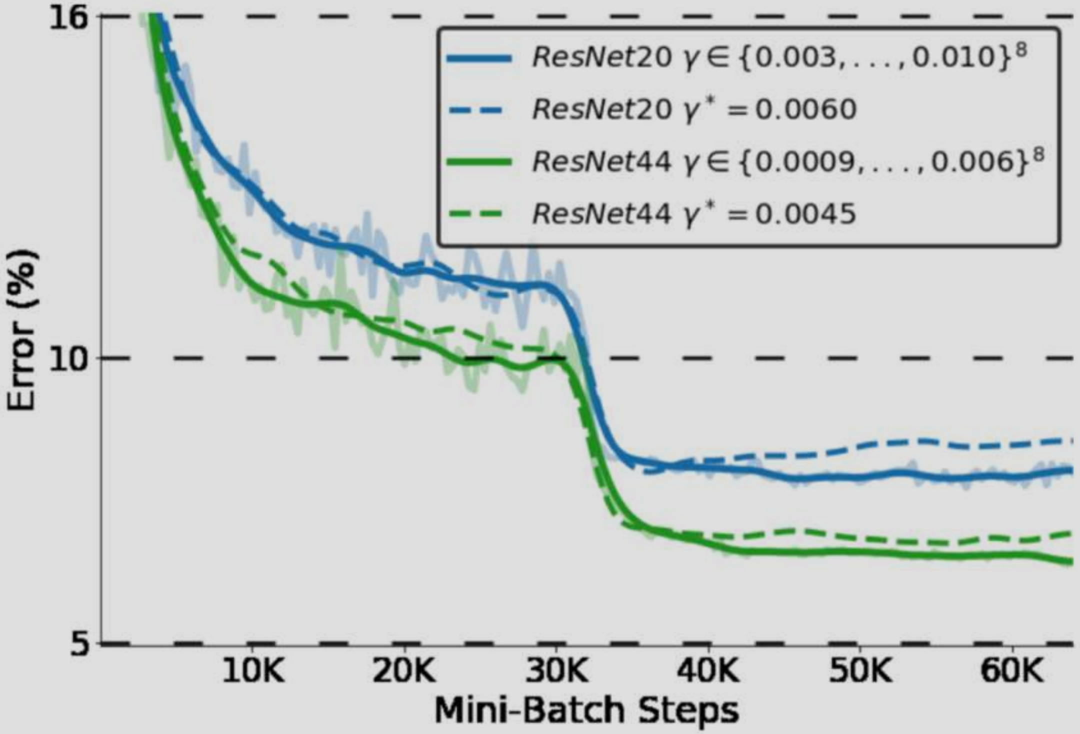
EMNIST

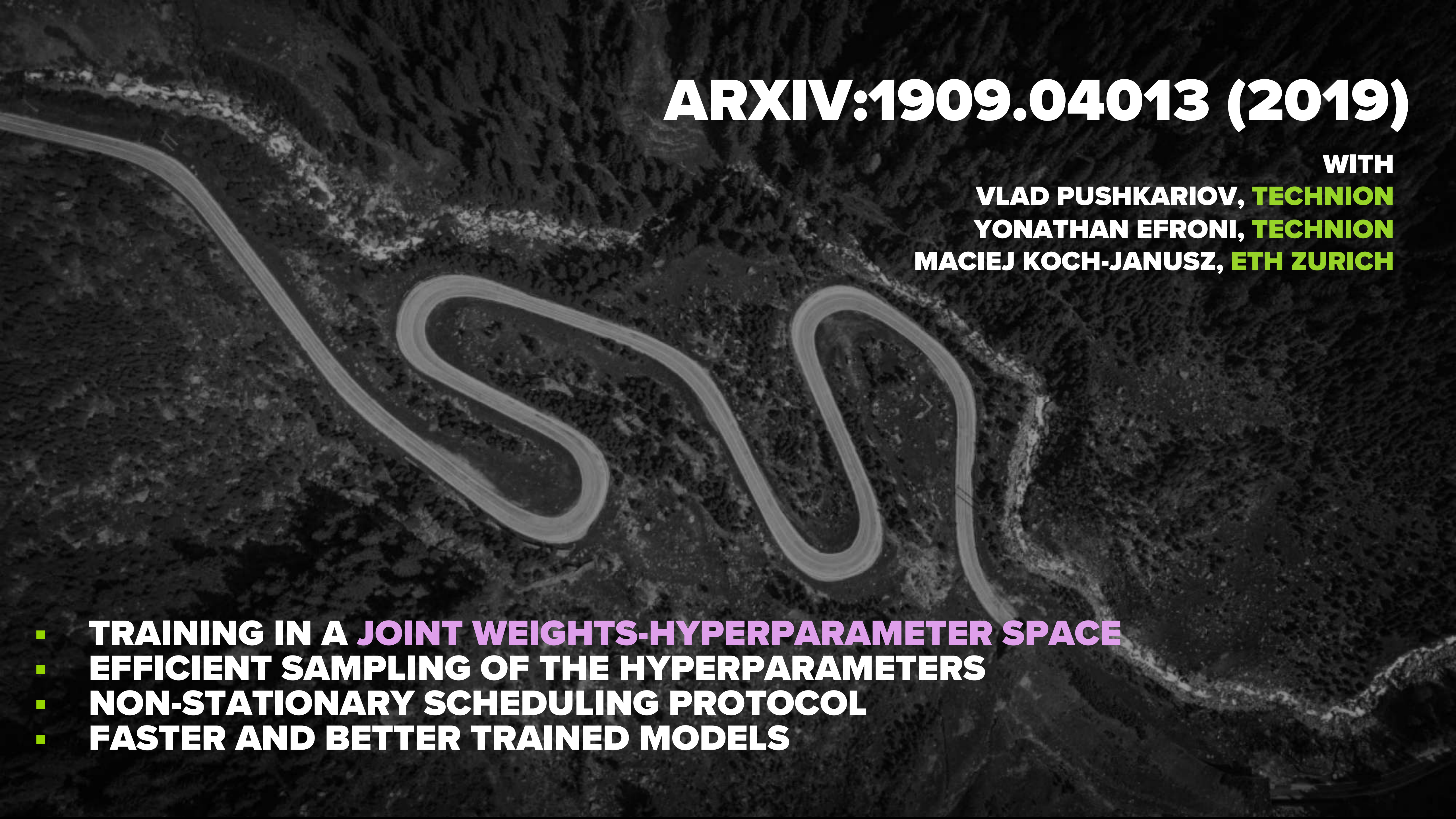
CIFAR10



RESNET/CIFAR10







ARXIV:1909.04013 (2019)

WITH
VLAD PUSHKARIOV, TECHNION
YONATHAN EFRONI, TECHNION
MACIEJ KOCH-JANUSZ, ETH ZURICH

- TRAINING IN A JOINT WEIGHTS-HYPERPARAMETER SPACE
- EFFICIENT SAMPLING OF THE HYPERPARAMETERS
- NON-STATIONARY SCHEDULING PROTOCOL
- FASTER AND BETTER TRAINED MODELS

THANKS FOR ATTENTION

Mykola Maksymenko

softserve