MetaGrad

Multiple Learning Rates in Online Learning

http://bitbucket.org/wmkoolen/metagrad

Tim van Erven

Universiteit Leiden

Wouter M. Koolen

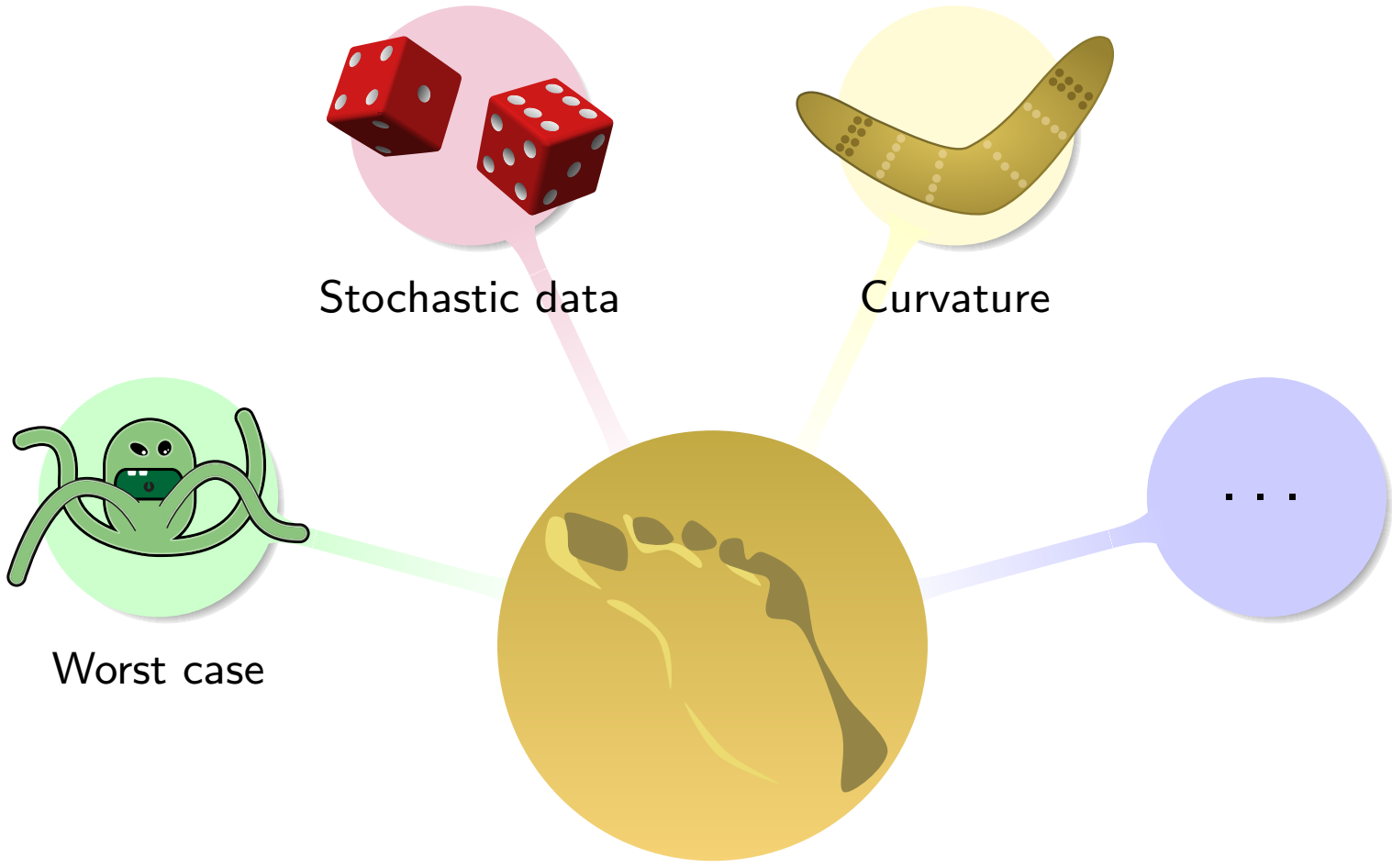
CWI

Centrum Wiskunde & Informatica

NIPS, Barcelona

Tuesday 6th December, 2016

In a Nutshell



Optimisation Pervasive in Machine Learning

$$\min_{\boldsymbol{w}} \sum_{t=1}^{T} f_t(\boldsymbol{w})$$

Optimisation Pervasive in Machine Learning

$$\min_{\boldsymbol{w}} \sum_{t=1}^{T} f_t(\boldsymbol{w})$$

Batch Training
(classification)

Optimisation Pervasive in Machine Learning

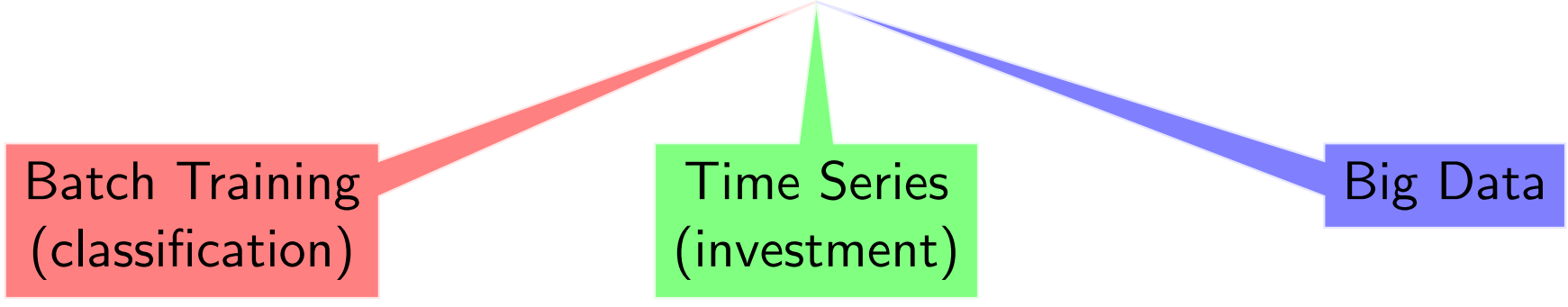
$$\min_{\boldsymbol{w}} \sum_{t=1}^{T} f_t(\boldsymbol{w})$$

Batch Training (classification)

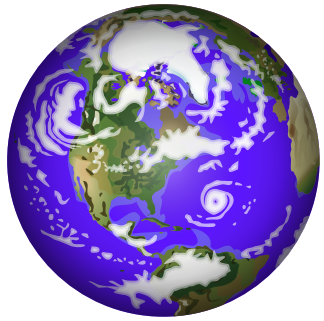
Time Series (investment)

Optimisation Pervasive in Machine Learning

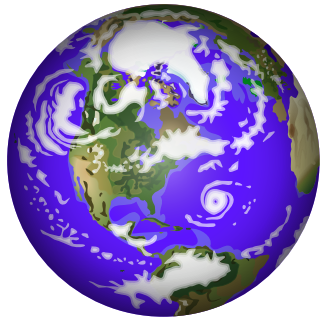
$$\min_{\boldsymbol{w}} \sum_{t=1}^{T} f_t(\boldsymbol{w})$$

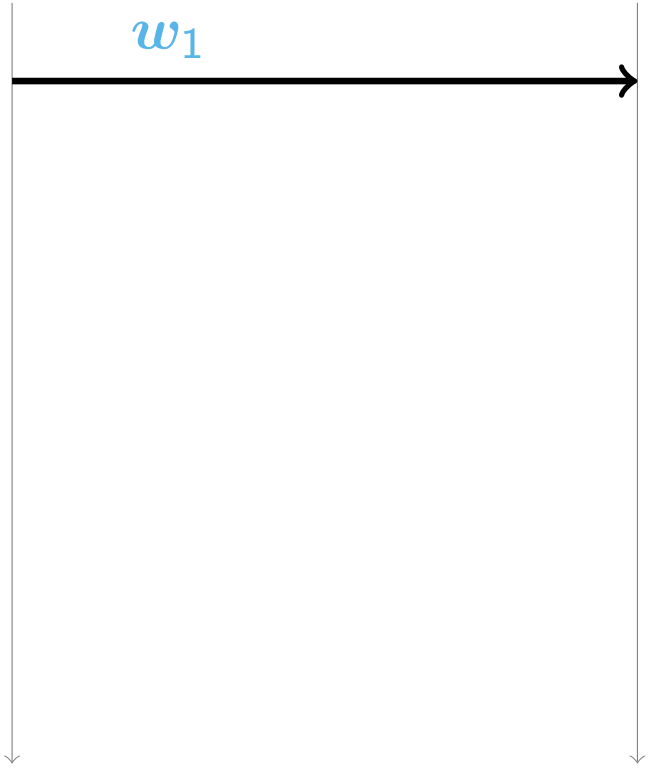


Online Convex Optimisation

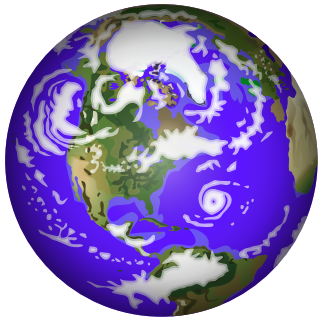


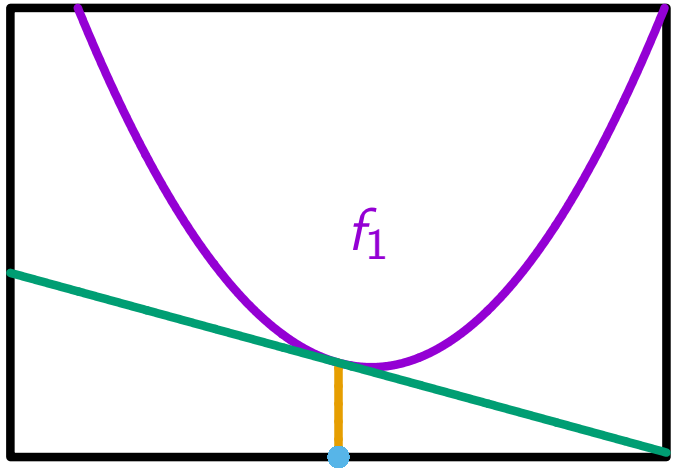
Online Convex Optimisation





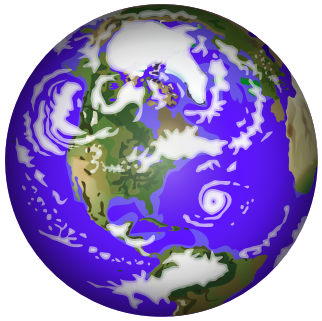
Online Convex Optimisation

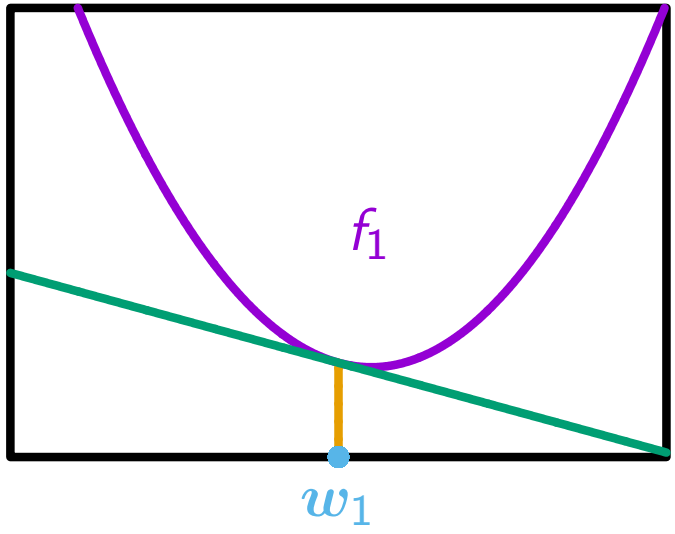




w_1

Online Convex Optimisation

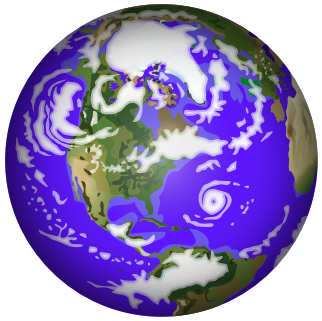


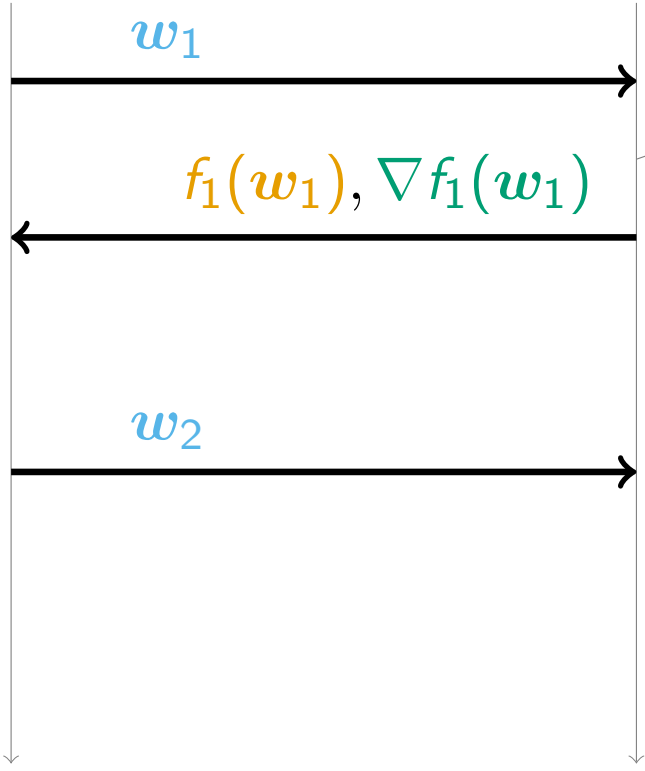


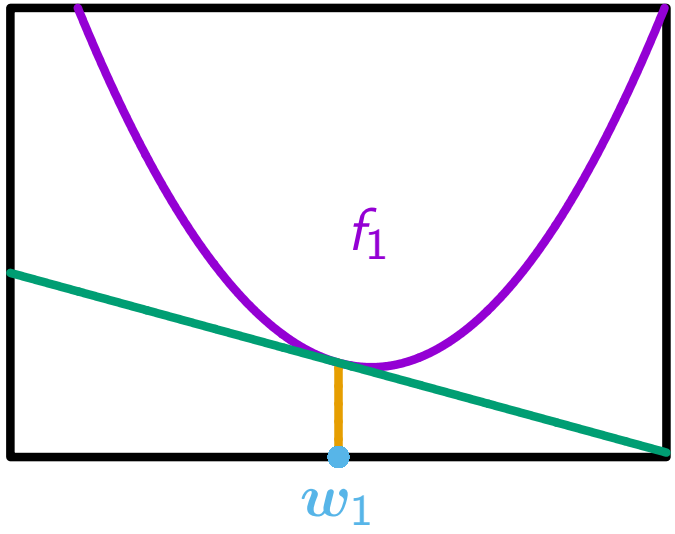
w_1

$f_1(w_1), \nabla f_1(w_1)$

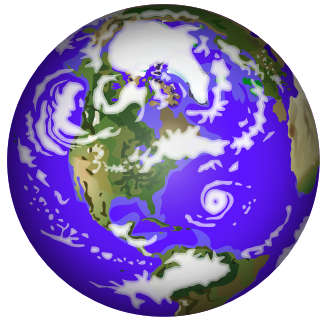
Online Convex Optimisation

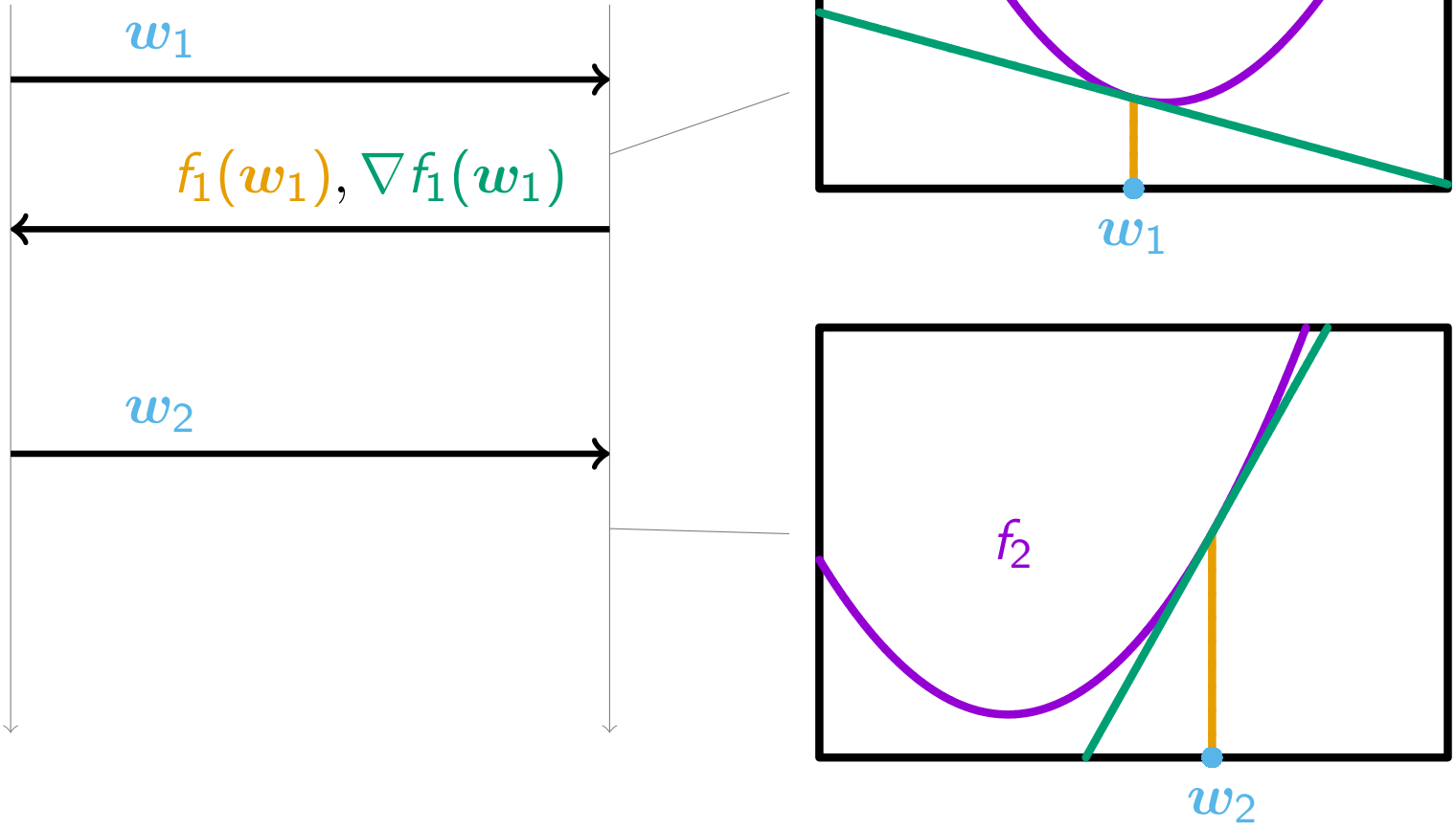




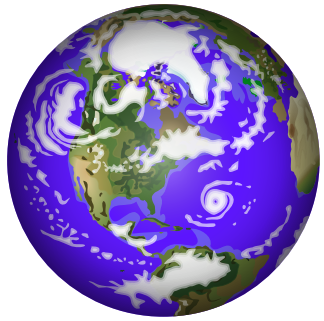


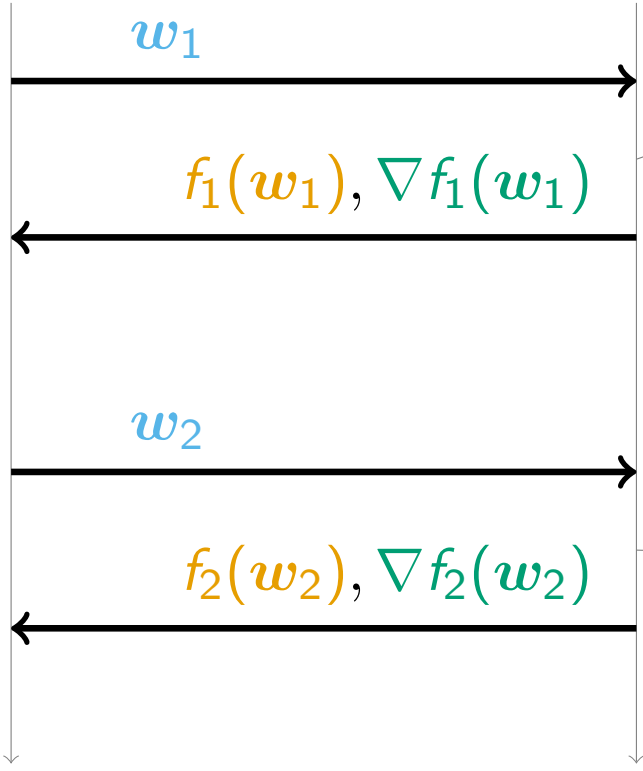
Online Convex Optimisation

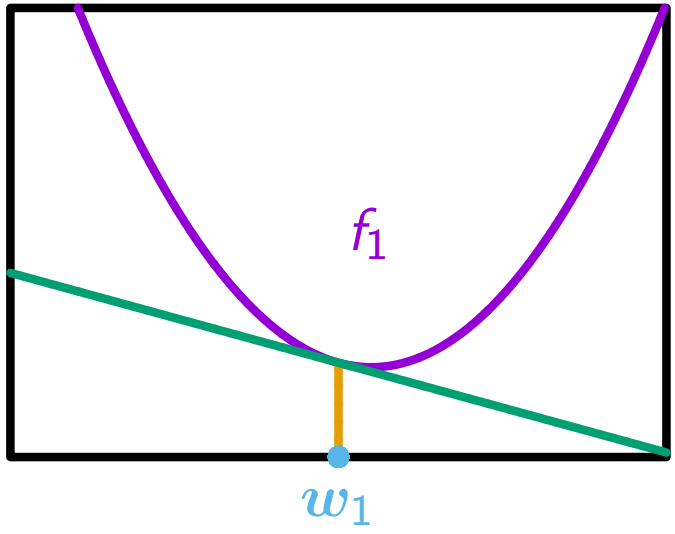


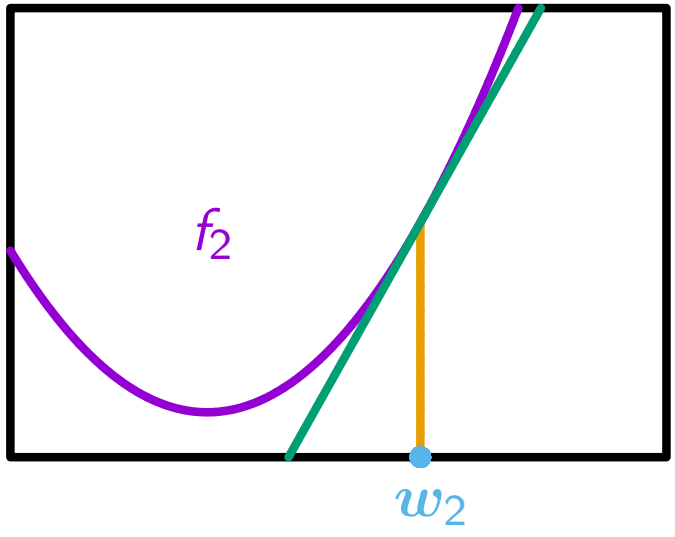


Online Convex Optimisation

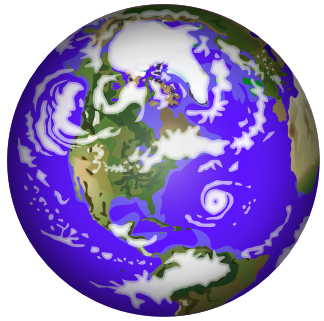


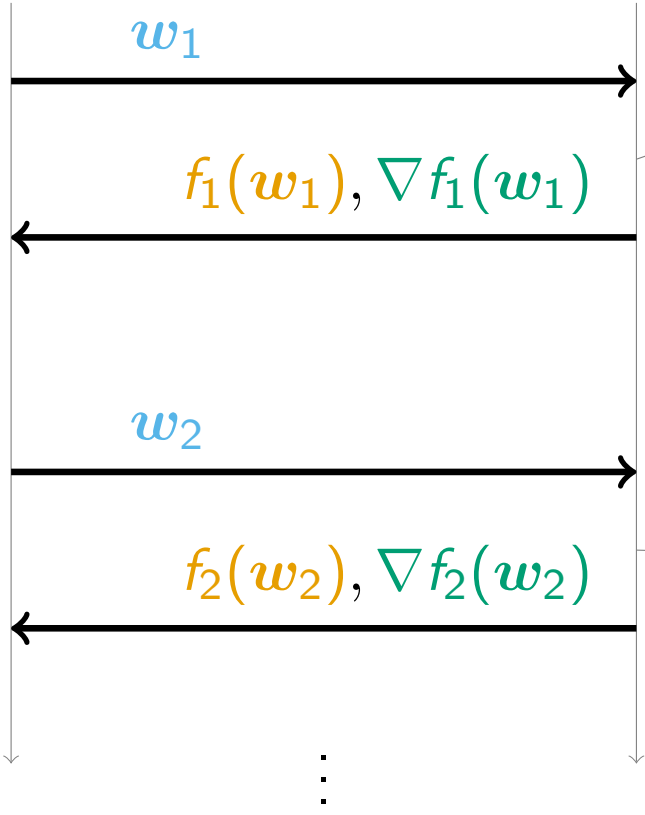


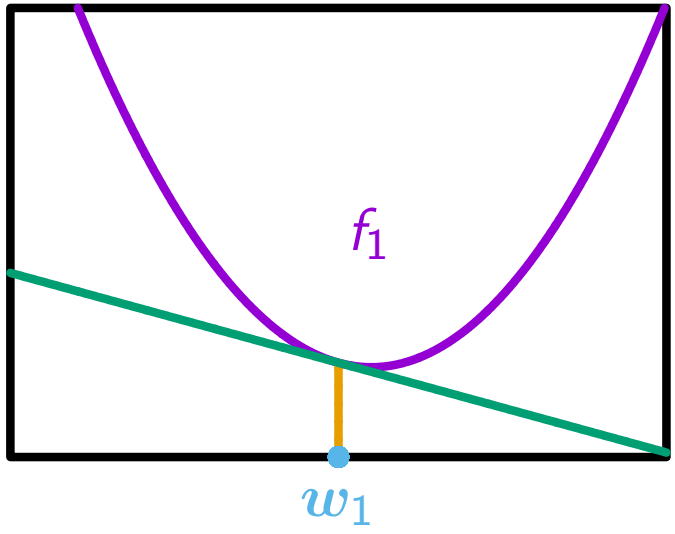


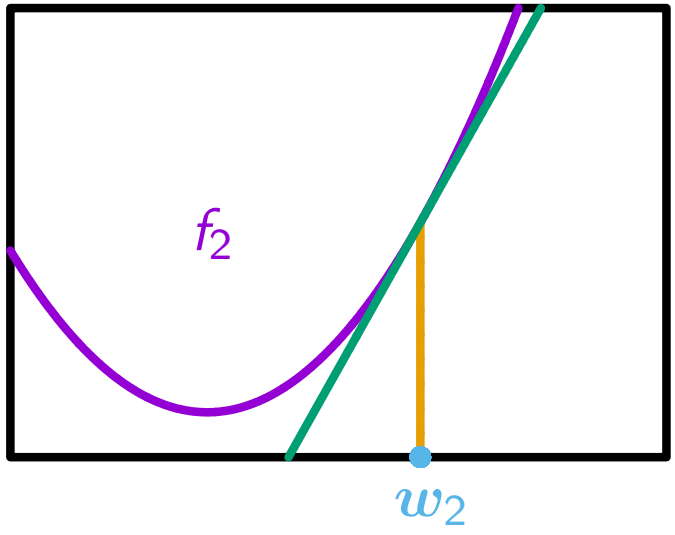


Online Convex Optimisation

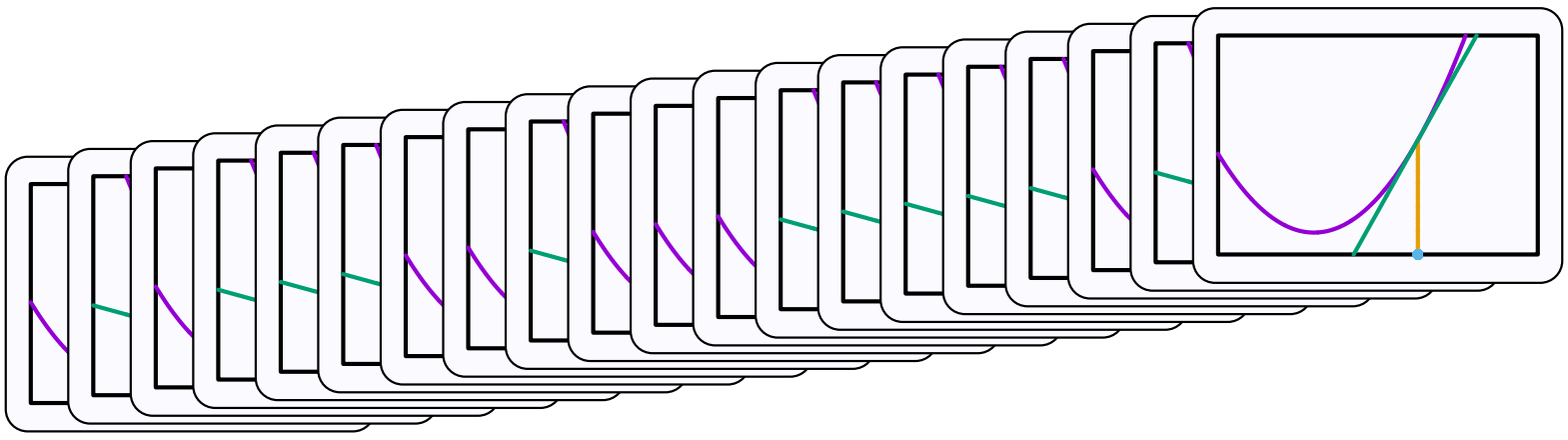








Objective



Definition (Regret)

$$R_T = \underbrace{\sum_{t=1}^{T} f_t(\boldsymbol{w}_t)}_{\text{Online loss}} - \underbrace{\min_{\boldsymbol{u}} \sum_{t=1}^{T} f_t(\boldsymbol{u})}_{\text{Optimal loss}}$$

Online Gradient Descent [Zinkevich, 2003]

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \nabla f_t(\boldsymbol{w}_t)$$

Online Gradient Descent [Zinkevich, 2003]

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \nabla f_t(\boldsymbol{w}_t)$$

Worst-case regret guarantee:

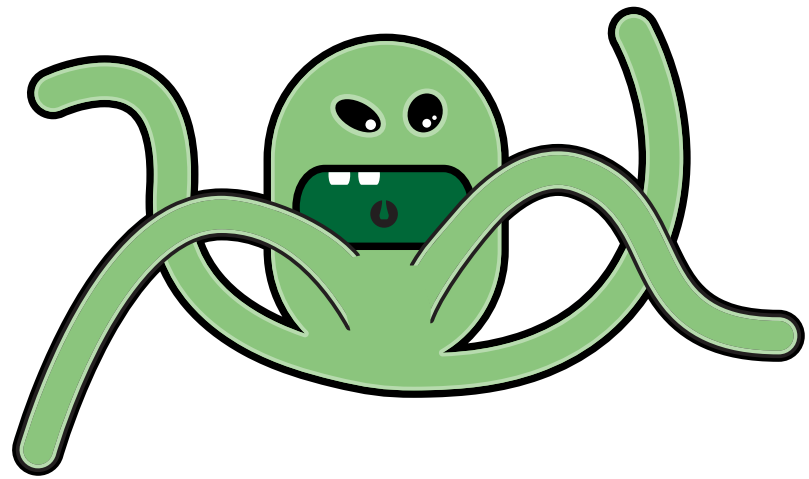
$$R_T = O\left(\sqrt{T}\right)$$

Online Gradient Descent [Zinkevich, 2003]

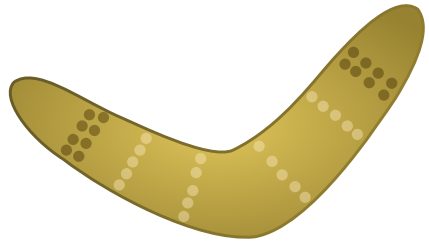
$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \nabla f_t(\boldsymbol{w}_t)$$

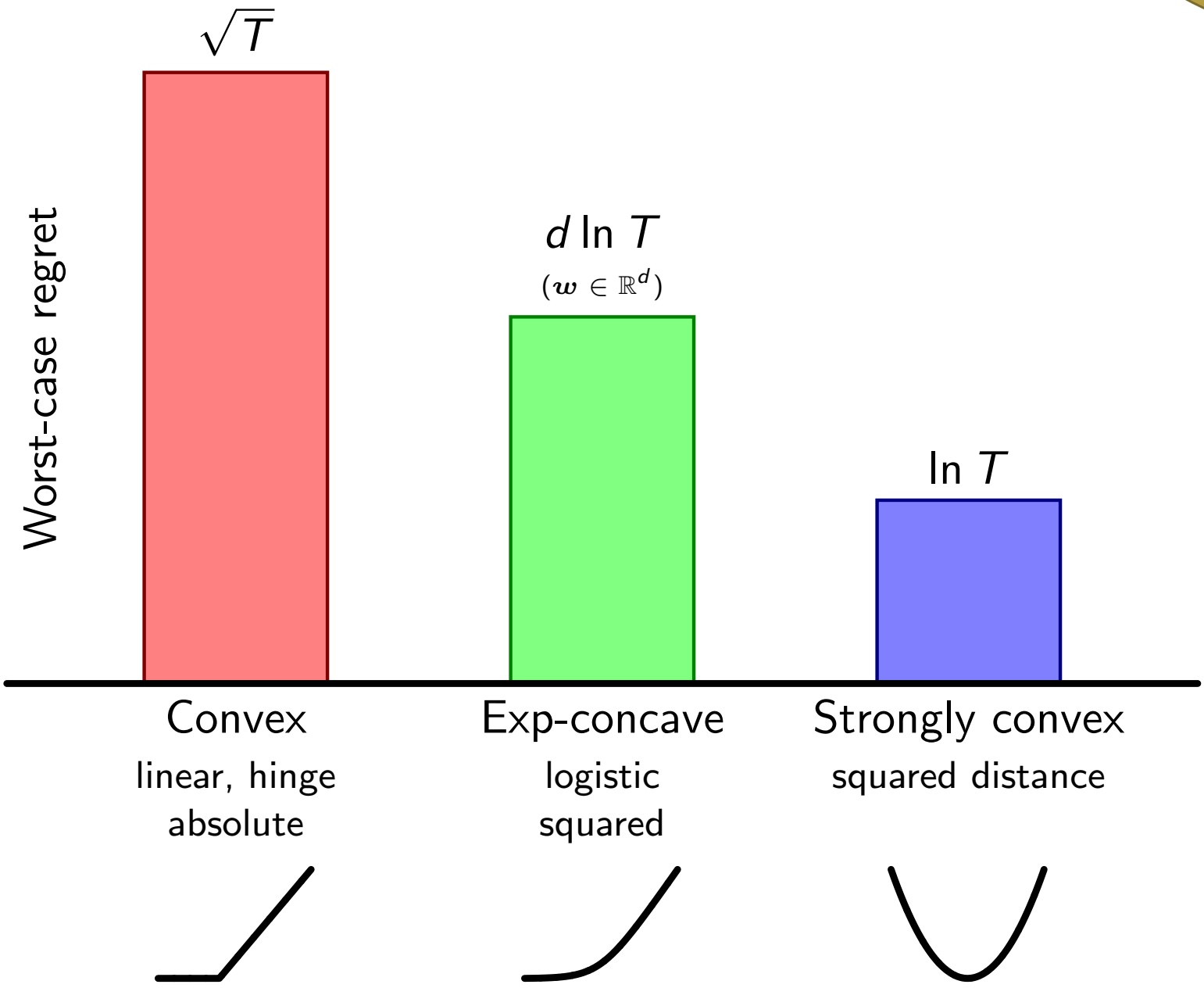
Worst-case regret guarantee:

$$R_T = O\left(\sqrt{T}\right)$$

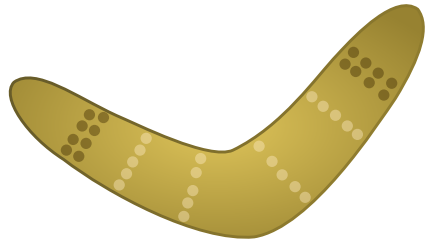


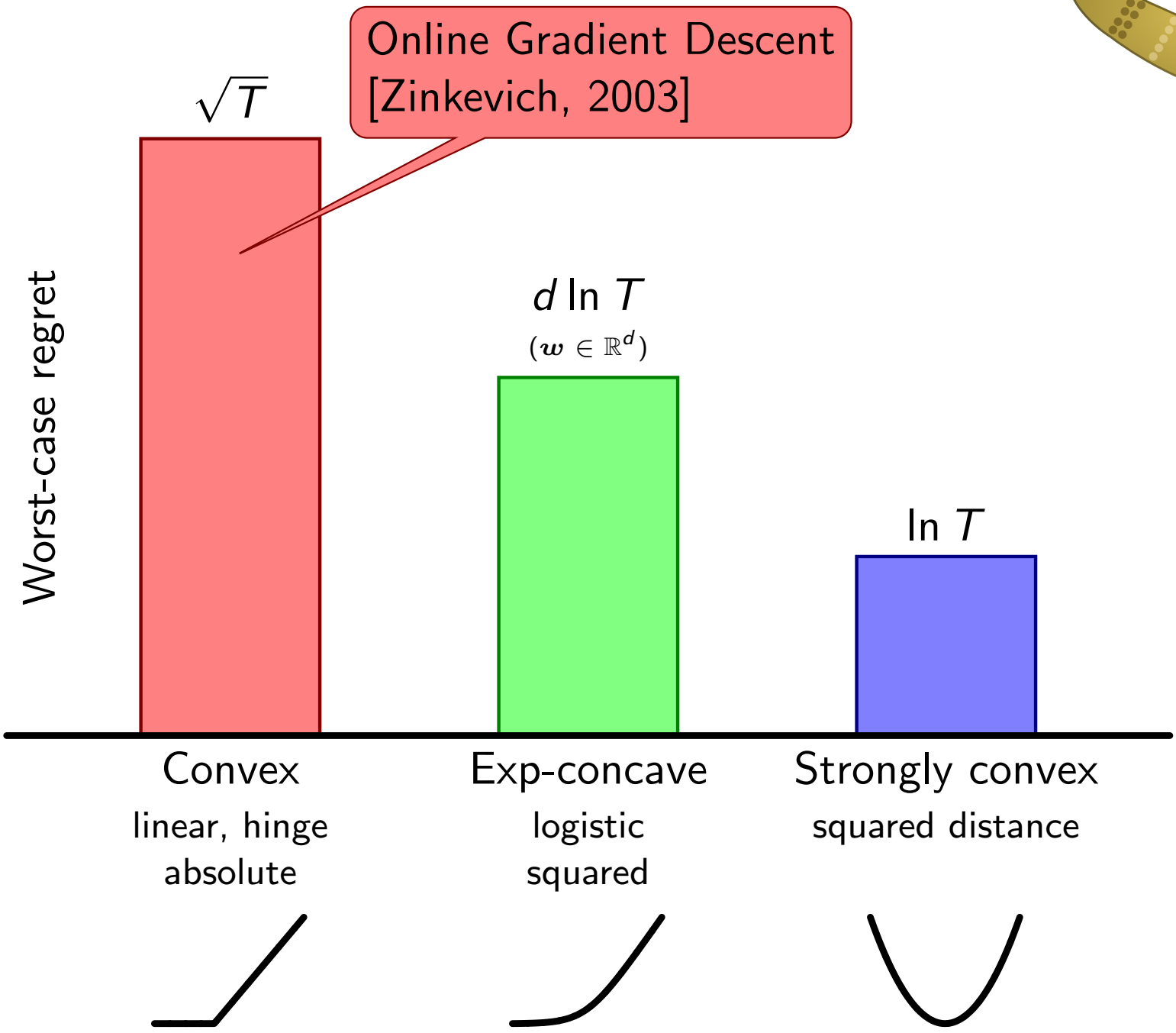
Loss Taxonomy $\sim$ Curvature



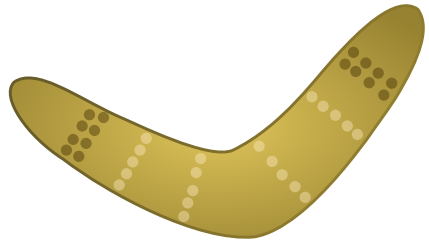


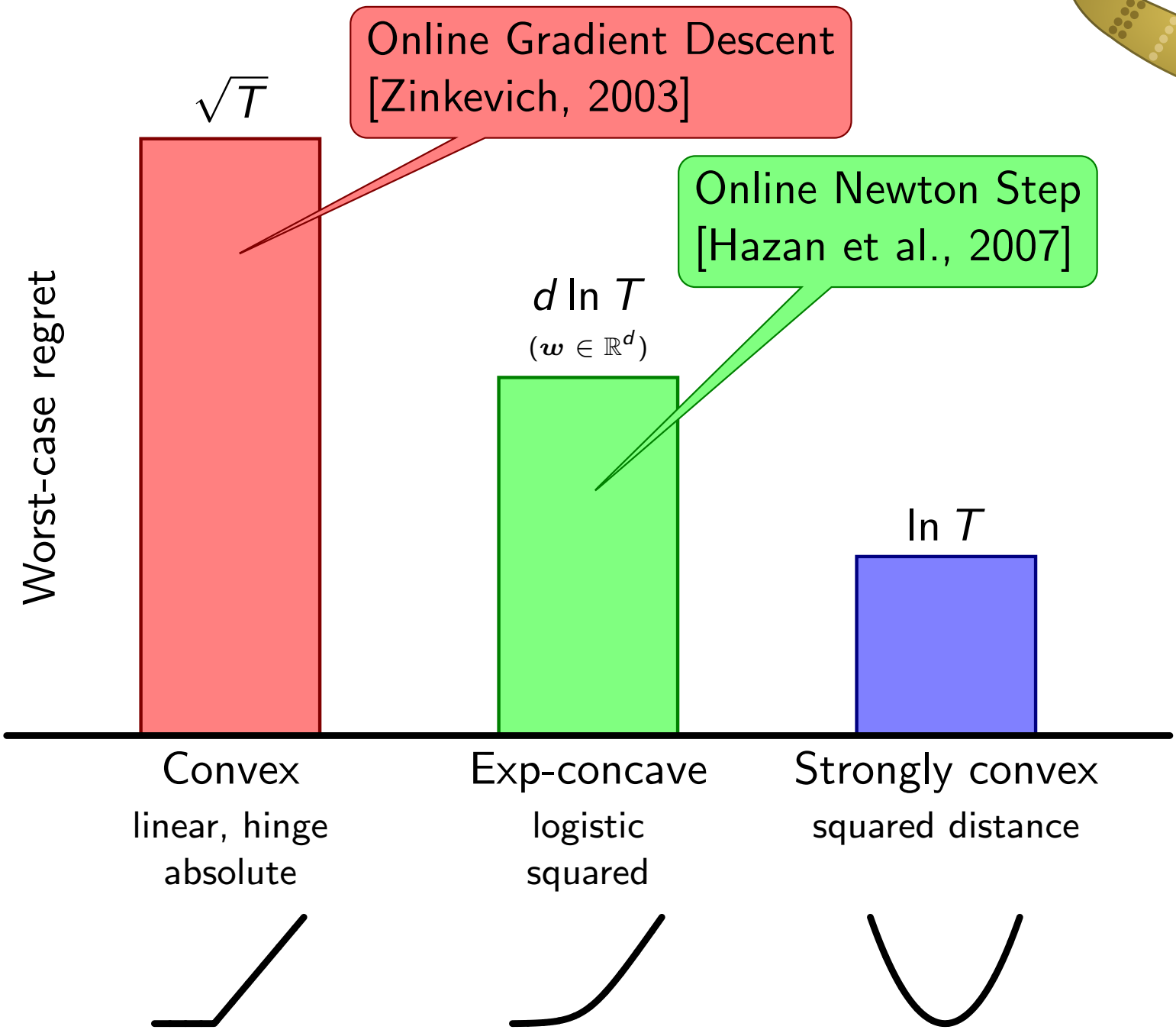
Loss Taxonomy ∼ Curvature



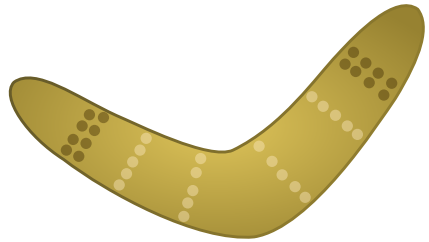


Loss Taxonomy ∼ Curvature





Loss Taxonomy ∼ Curvature



Big Questions

Can we make **adaptive** methods for **online convex optimisation** that are

- **worst-case safe**
- exploit **curvature** automatically
- computationally **efficient**

Big Questions

Can we make **adaptive** methods for **online convex optimisation** that are

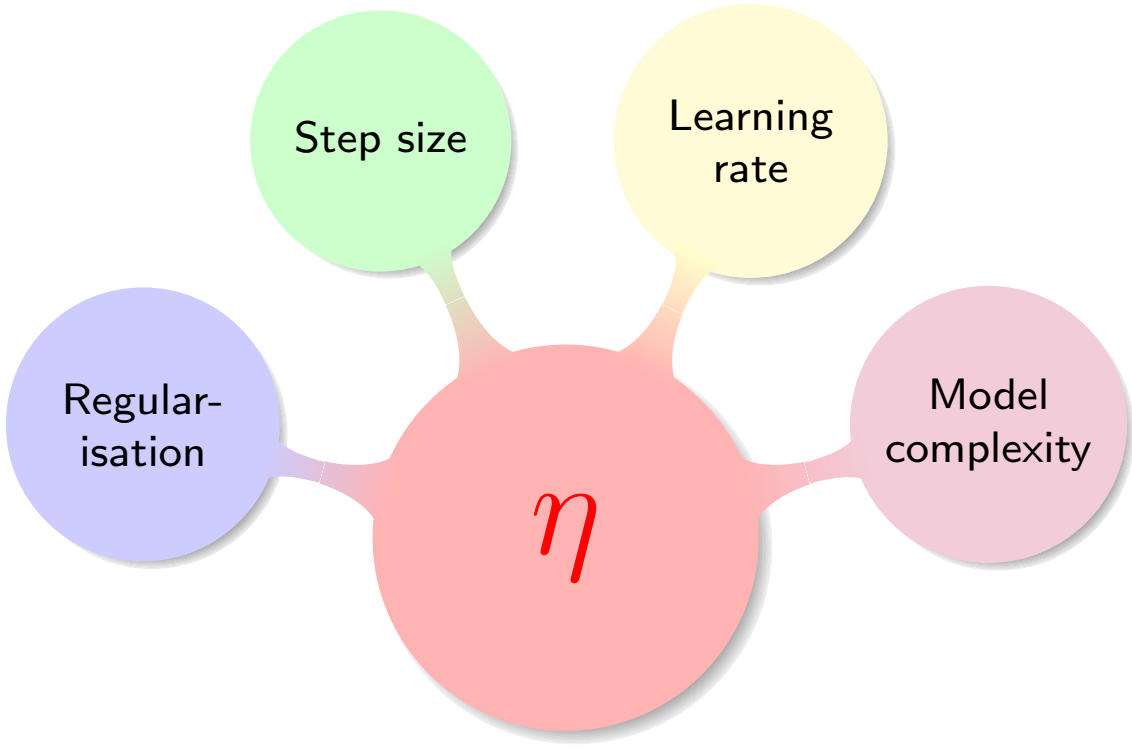
- **worst-case safe**
- exploit **curvature** automatically
- computationally **efficient**

And can we adapt to other important regimes?

- **Mixed** or **in-between** cases?
- **Stochastic** data? Bandits [Seldin and Slivkins, 2014]
- Absence of **curvature**? Experts [Koolen and Van Erven, 2015]

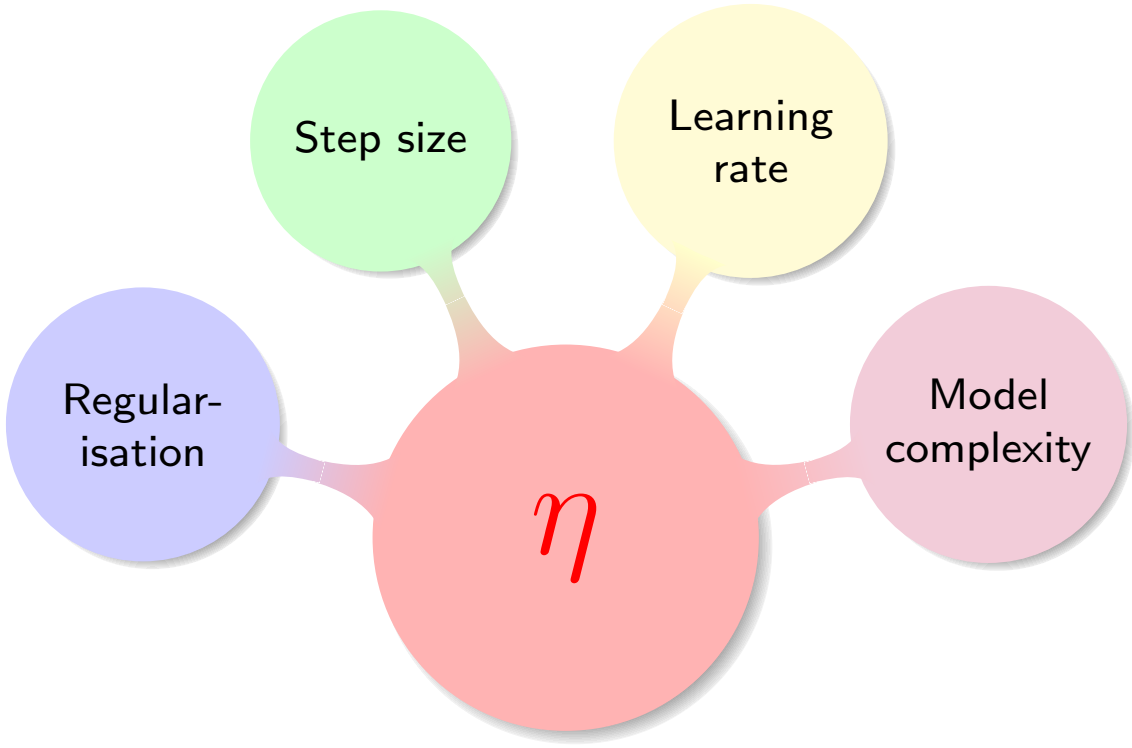
Main Idea

For every optimisation algorithm tuning is **crucial**.



Main Idea

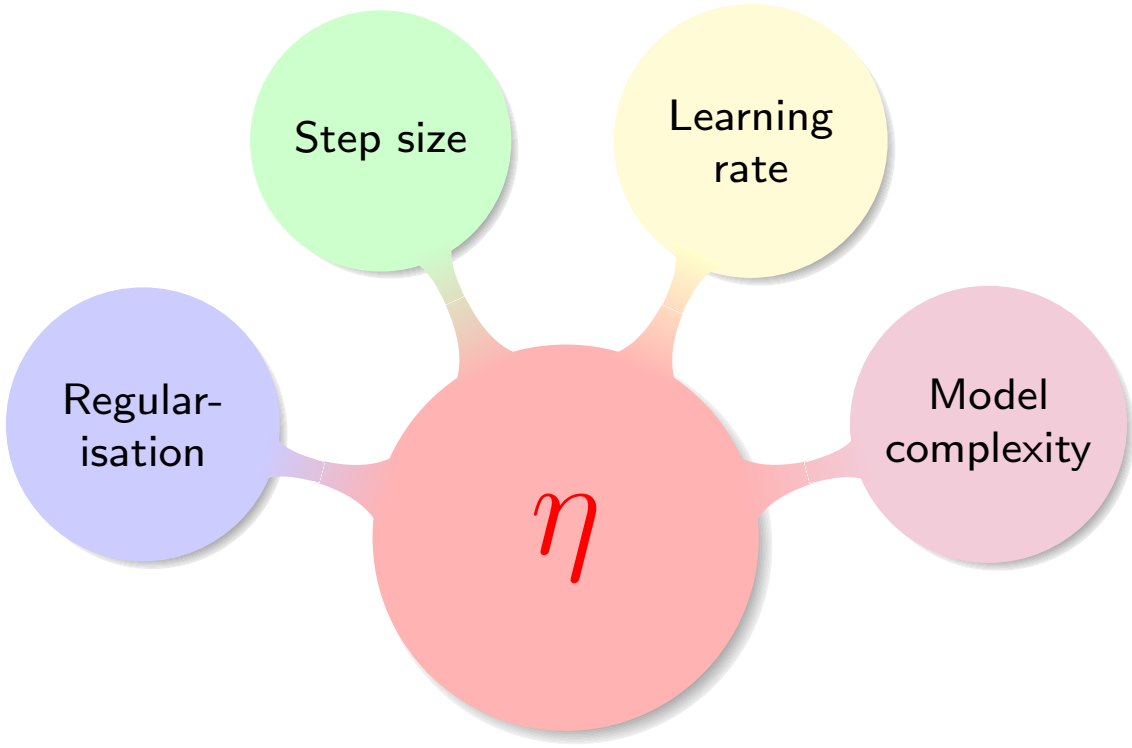
For every optimisation algorithm tuning is **crucial**.



So let's **learn** optimal tuning from **data**.

Main Idea

For every optimisation algorithm tuning is **crucial**.

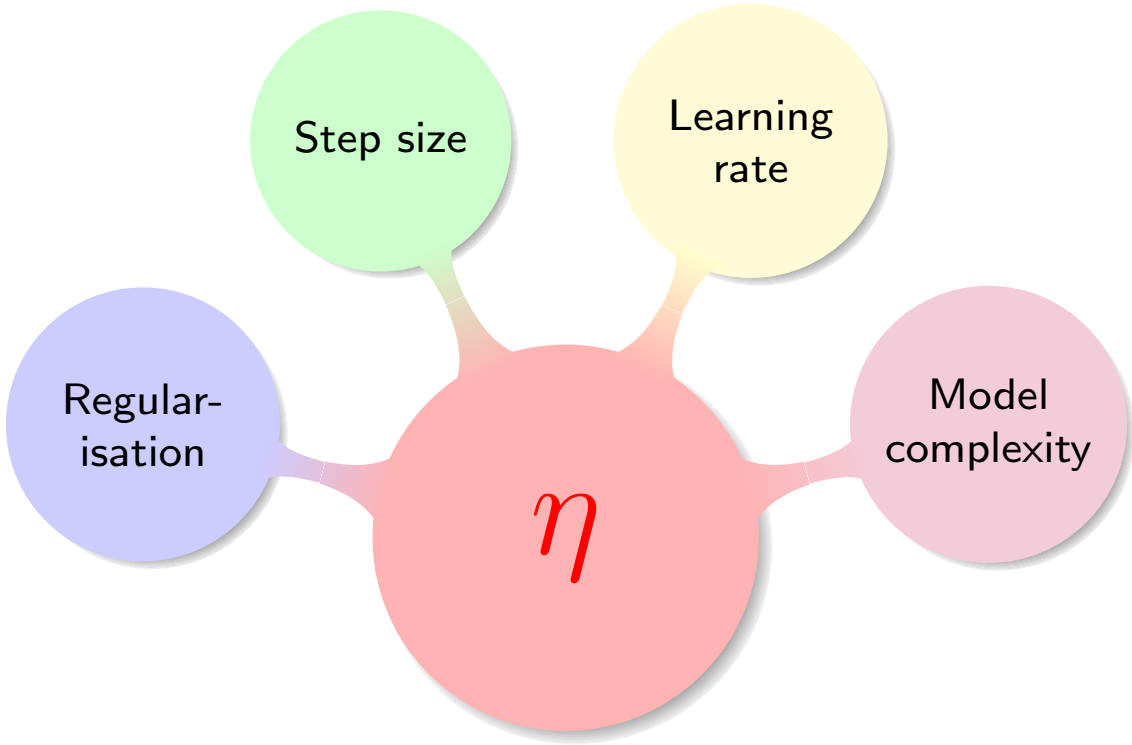


So let's **learn** optimal tuning from **data**.

Key obstacle: avoid learning η at **slow rate** itself.

Main Idea

For every optimisation algorithm tuning is **crucial**.



So let's **learn** optimal tuning from **data**.

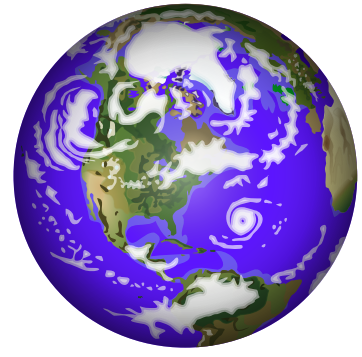
Key obstacle: avoid learning η at **slow rate** itself.

Breakthrough: Multiple Eta Gradient algorithm (MetaGrad)

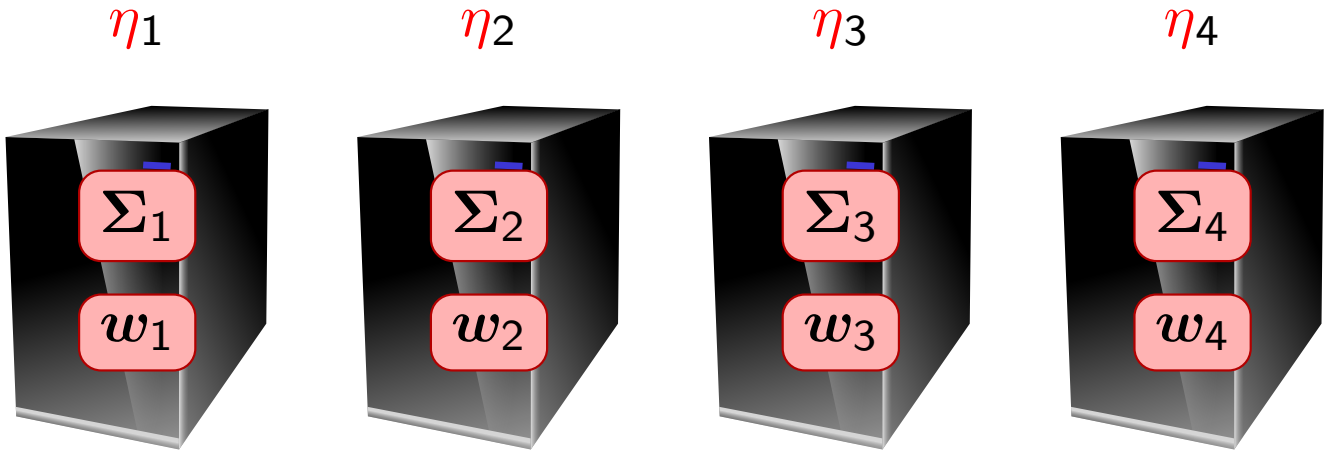
MetaGrad Algorithm

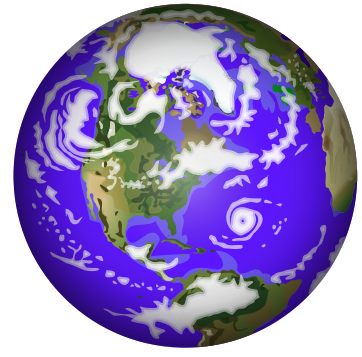
η_1 η_2 η_3 η_4

$\cdots \underbrace{\ln(T)}_{\leq 16}$

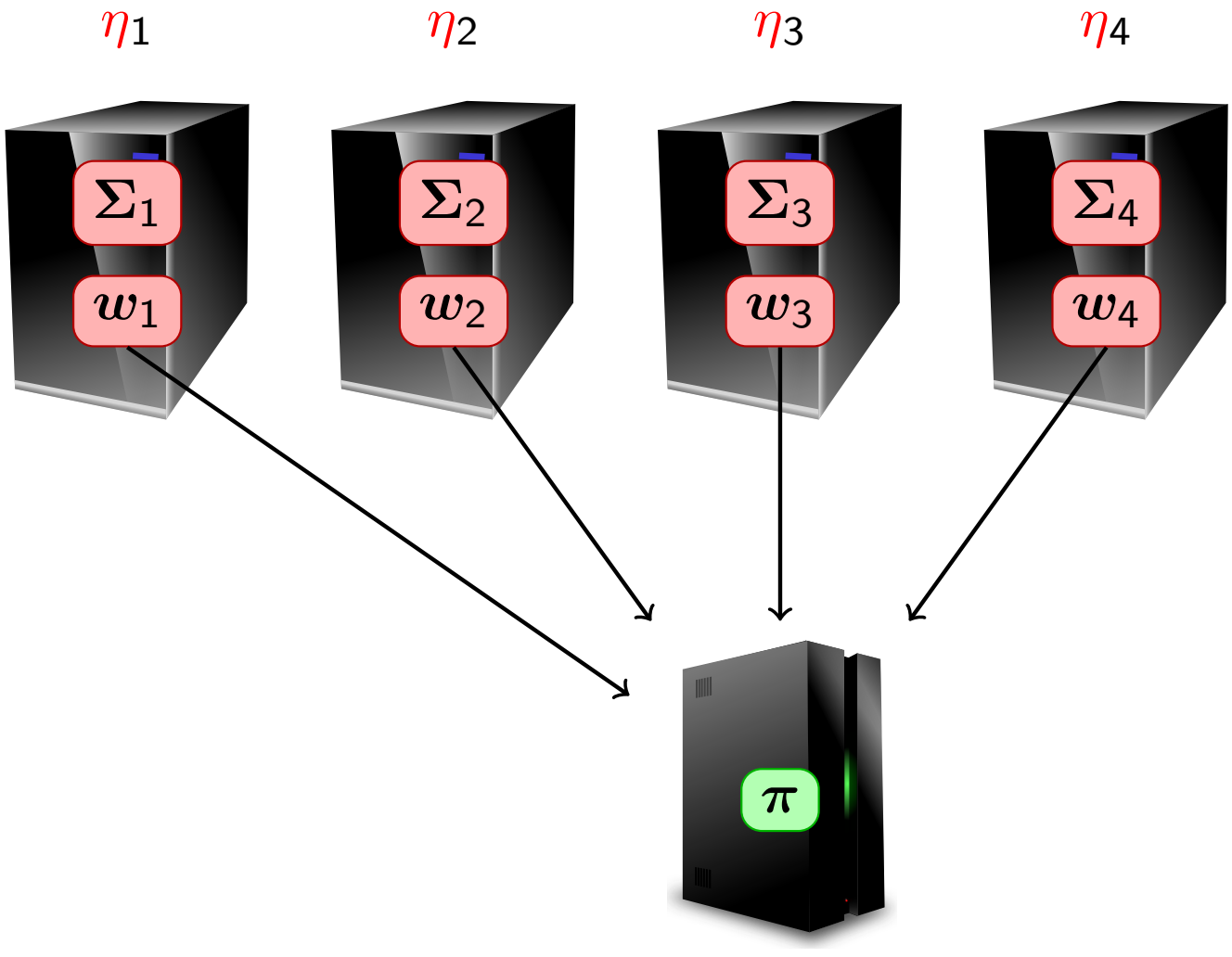


MetaGrad Algorithm

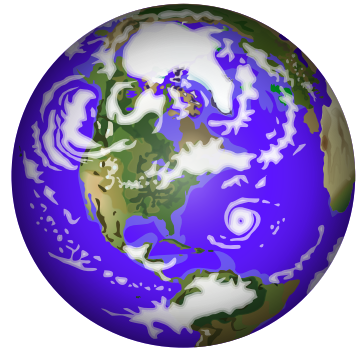




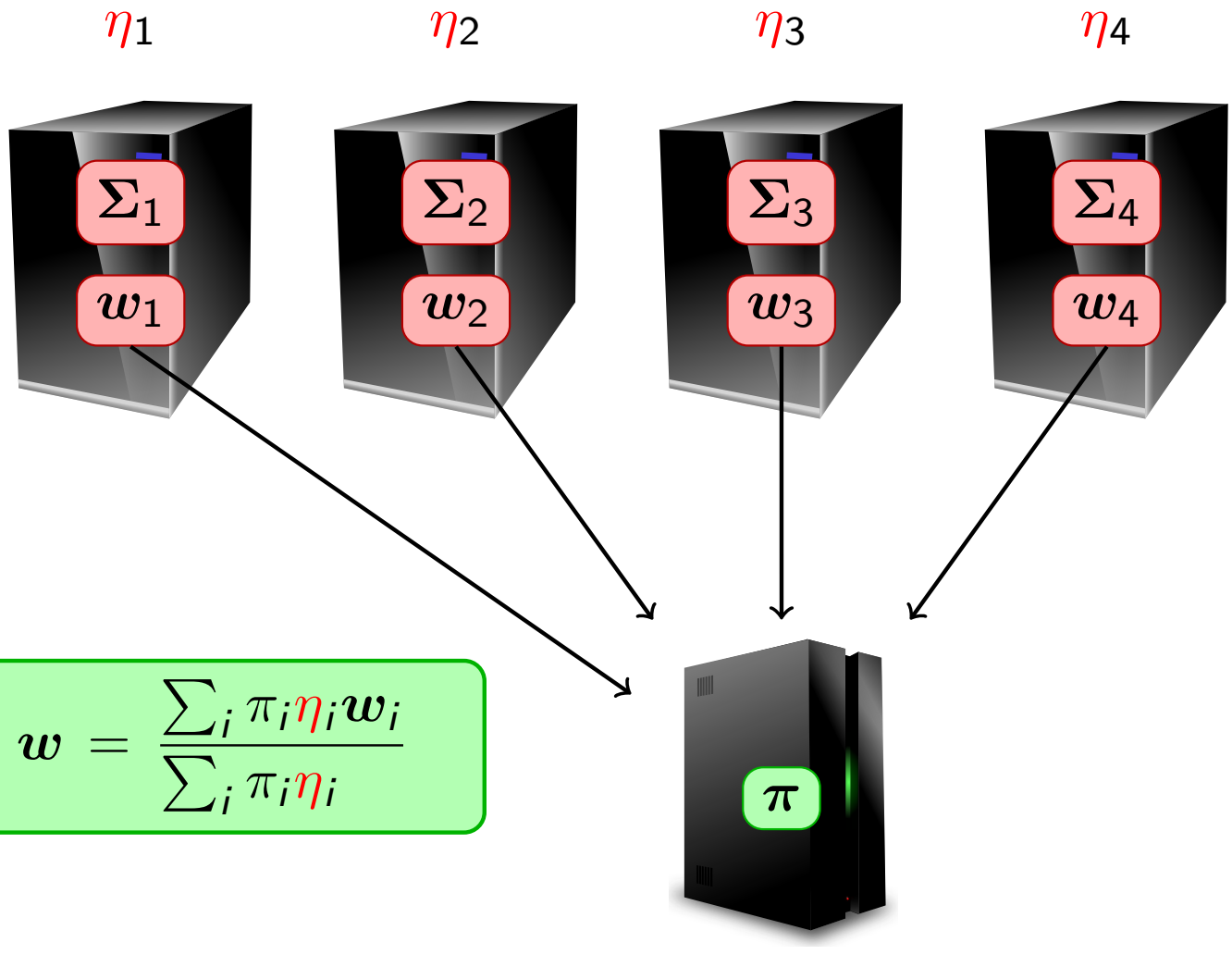
MetaGrad Algorithm



$$\cdots \underbrace{\ln(T)}_{\leq 16}$$

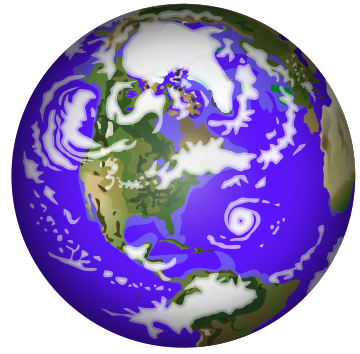


MetaGrad Algorithm

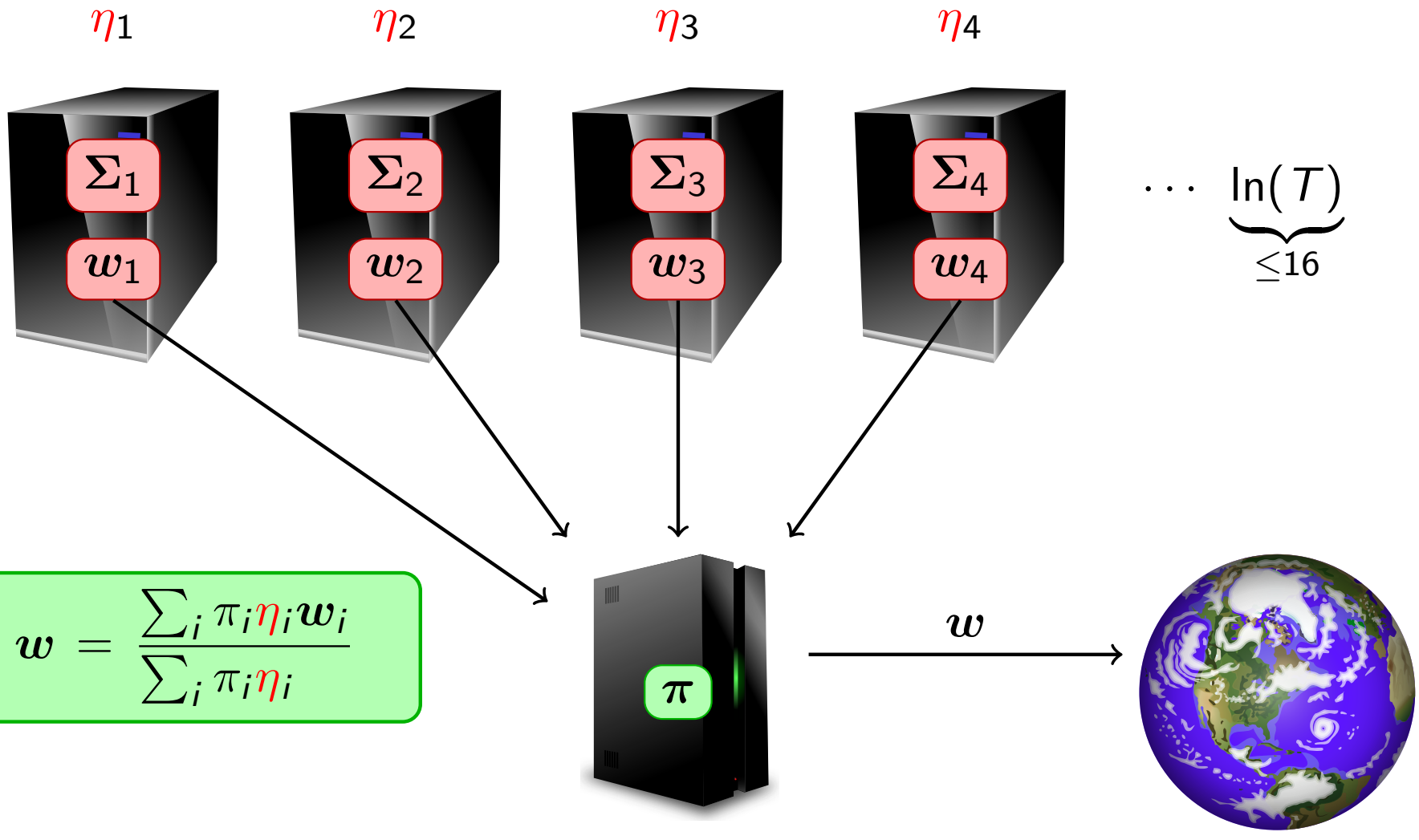


$$\cdots \underbrace{\ln(T)}_{\leq 16}$$

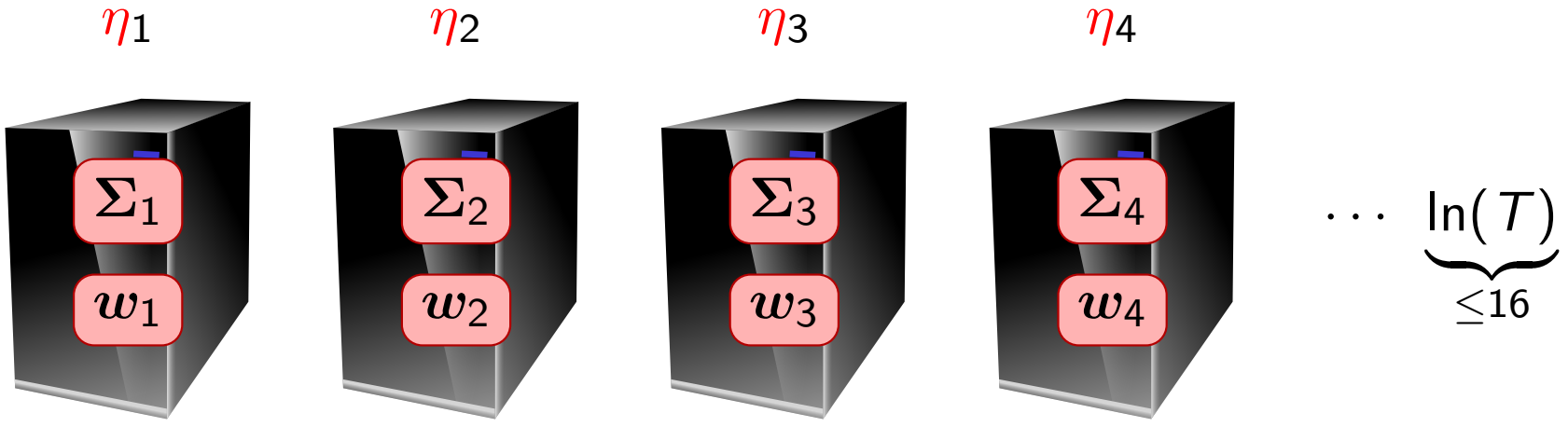
$$w = \frac{\sum_i \pi_i \eta_i w_i}{\sum_i \pi_i \eta_i}$$



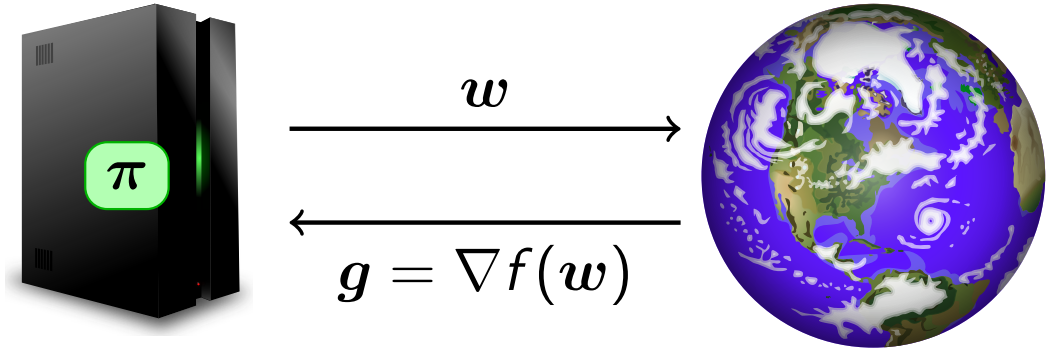
MetaGrad Algorithm



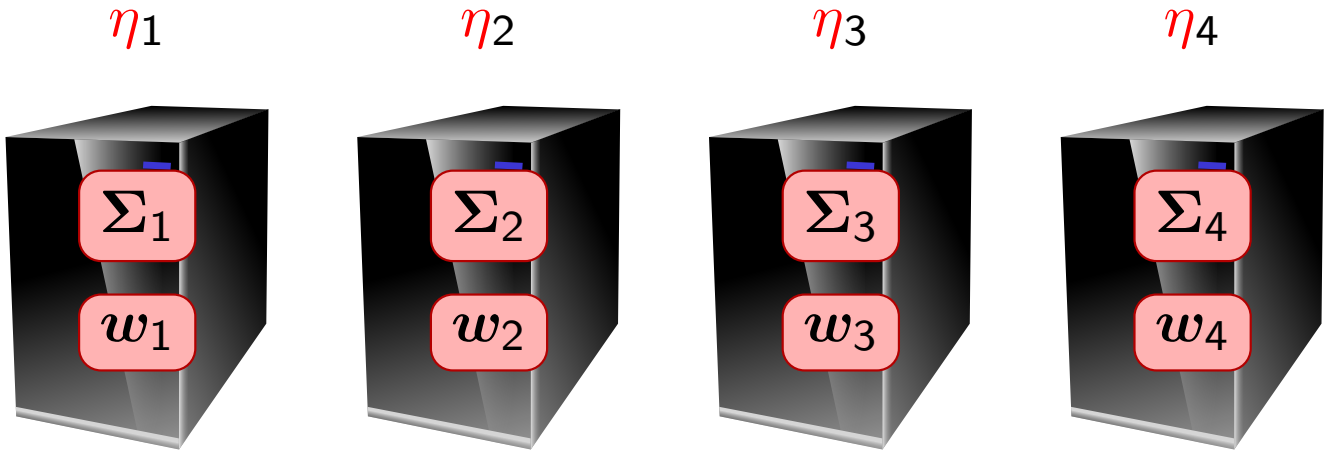
MetaGrad Algorithm



$$w = \frac{\sum_i \pi_i \eta_i w_i}{\sum_i \pi_i \eta_i}$$



MetaGrad Algorithm

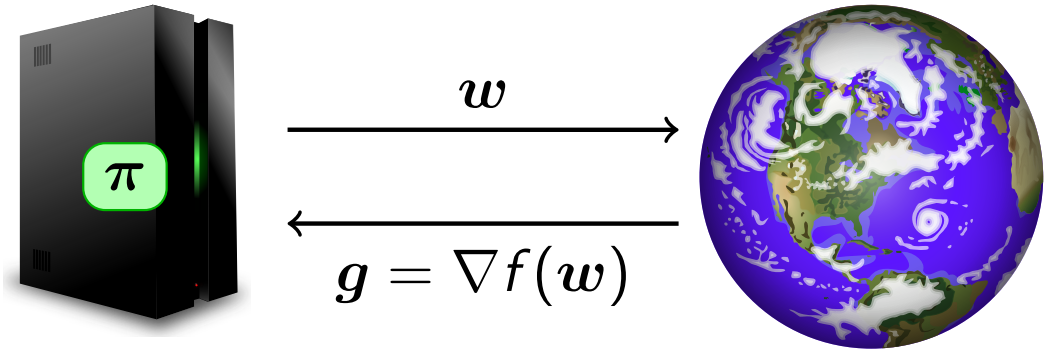


$$w = \frac{\sum_i \pi_i \eta_i w_i}{\sum_i \pi_i \eta_i}$$

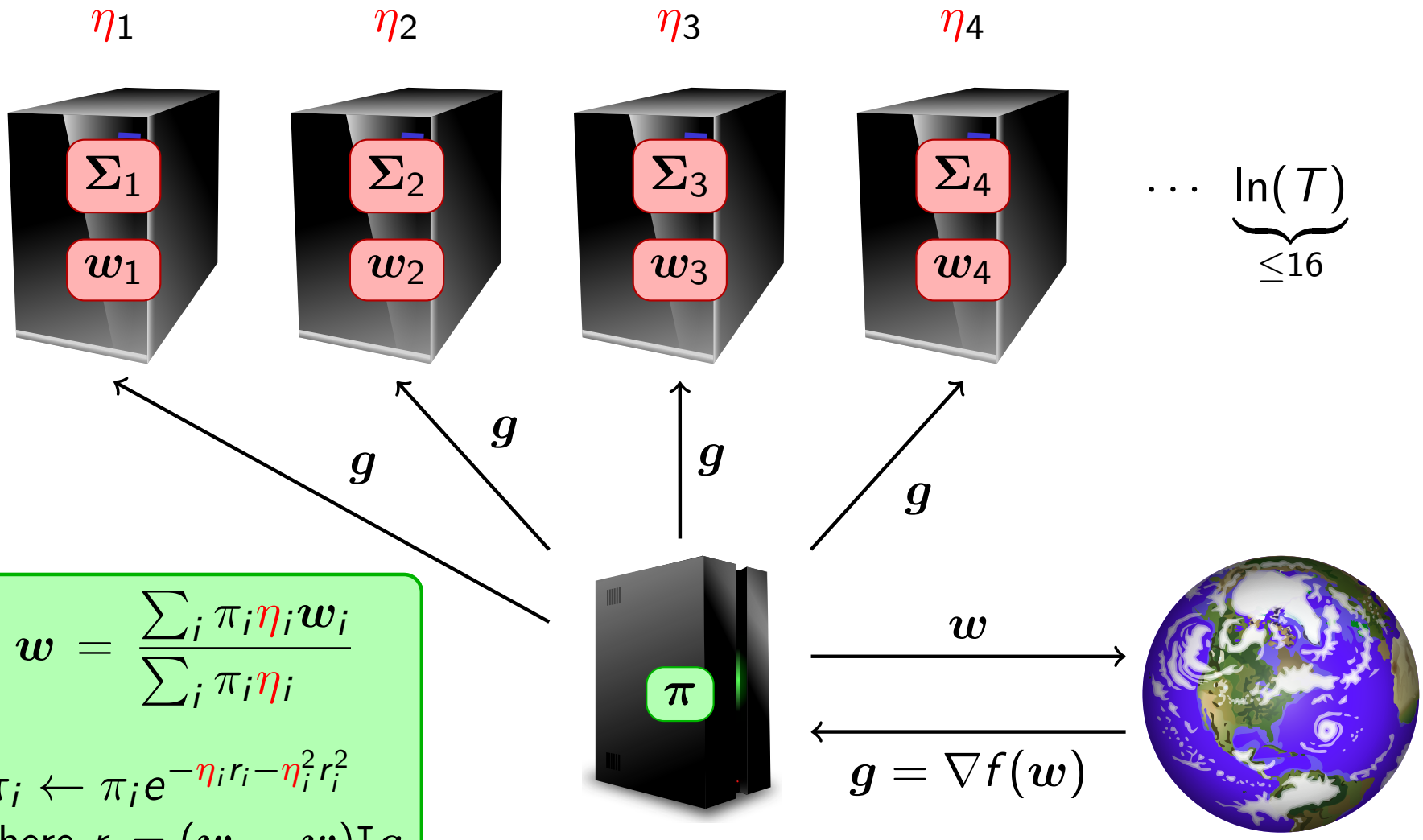
$$\pi_i \leftarrow \pi_i e^{-\eta_i r_i - \eta_i^2 r_i^2}$$

where $r_i = (w_i - w)^\intercal g$

Tilted Exponential Weights



MetaGrad Algorithm



$$w = \frac{\sum_i \pi_i \eta_i w_i}{\sum_i \pi_i \eta_i}$$

$$\pi_i \leftarrow \pi_i e^{-\eta_i r_i - \eta_i^2 r_i^2}$$

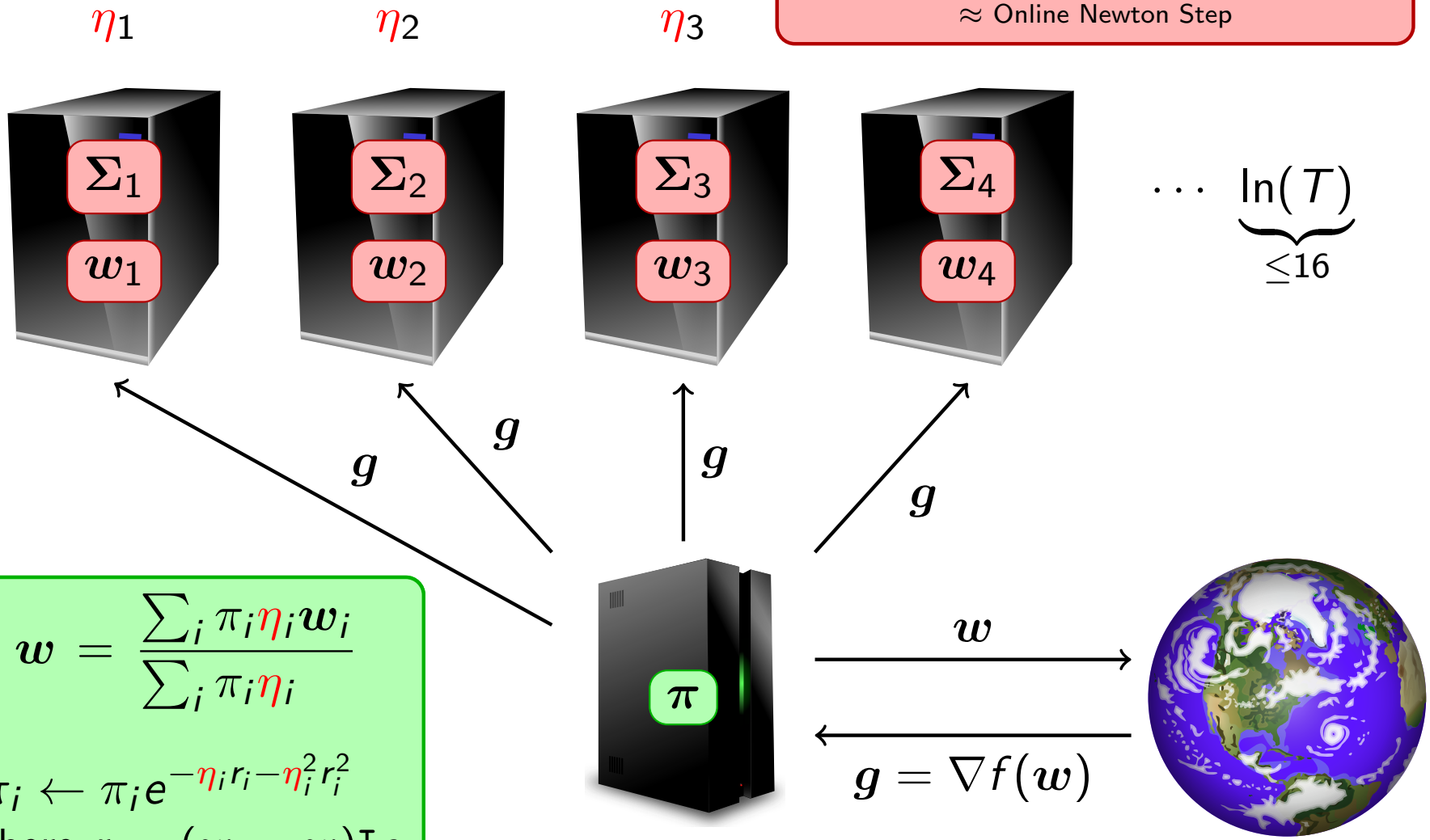
where $r_i = (w_i - w)^\intercal g$

Tilted Exponential Weights

MetaGrad Algorithm

$$\Sigma_i \leftarrow (\Sigma_i^{-1} + 2\eta_i^2 g g^\intercal)^{-1}$$
$$w_i \leftarrow w_i - \eta_i \Sigma_i g \,(1 + 2\eta_i r_i)$$

$\approx$ Online Newton Step

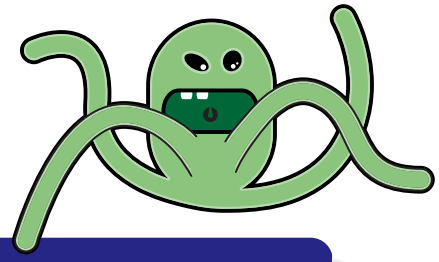


$$w = \frac{\sum_i \pi_i \eta_i w_i}{\sum_i \pi_i \eta_i}$$

$$\pi_i \leftarrow \pi_i e^{-\eta_i r_i - \eta_i^2 r_i^2}$$
where $r_i = (w_i - w)^\intercal g$

Tilted Exponential Weights

Second-order Regret Bound



Theorem

The regret of MetaGrad is bounded by

$$R_T \;=\; O\left(\min\left\{\sqrt{T}, \sqrt{V_T\, d \ln T}\right\}\right),$$

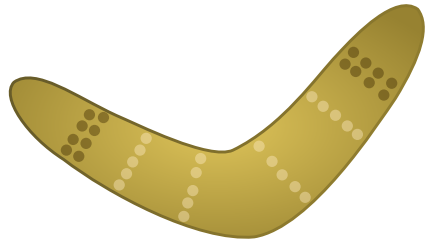
where

$$V_T \;=\; \sum_{t=1}^{T}\left((\boldsymbol{w}_t - \boldsymbol{u}^*)^\intercal \nabla f_t(\boldsymbol{w}_t)\right)^2$$

measures **variance** *compared to the* **offline optimum** $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} \sum_{t=1}^{T} f_t(\boldsymbol{u})$

Note: Optimal tuning depends on unknown optimum $\boldsymbol{u}^*$.

MetaGrad Adapts to Curvature



MetaGrad regret bound:

$$R_T \;=\; O\left(\sqrt{V_T \, d \ln T}\right)$$

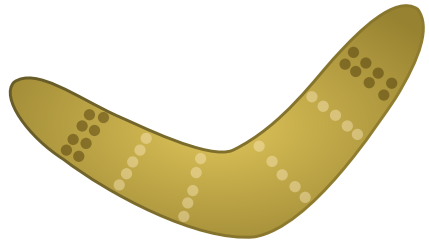
Corollary

For α-**exp-concave** *or* α-**strongly convex** *losses, MetaGrad ensures*

$$R_T \;=\; O\left(d \ln T\right)$$

without knowing α.

MetaGrad Adapts to Curvature



MetaGrad regret bound:

$$R_T \;=\; O\left(\sqrt{V_T \, d \ln T}\right)$$

Corollary

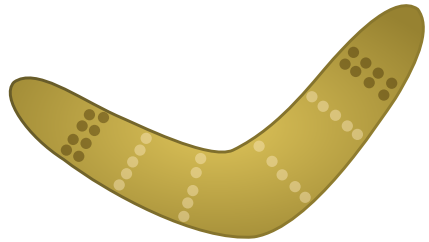
*For α-**exp-concave** or α-**strongly convex** losses, MetaGrad ensures*

$$R_T \;=\; O\left(d \ln T\right)$$

without knowing α.

Same result for fixed $f_t = f$ (classical optimisation) even without curvature via **derivative condition**.

MetaGrad Adapts to Curvature



MetaGrad regret bound:

$$R_T \;=\; O\left(\sqrt{V_T\, d \ln T}\right)$$

Corollary

For α-**exp-concave** *or* α-**strongly convex** *losses, MetaGrad ensures*

$$R_T \;=\; O\left(d \ln T\right)$$

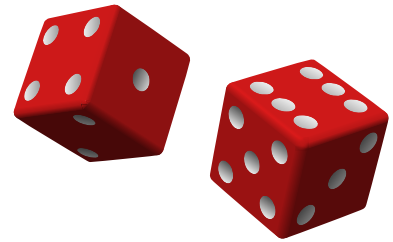
without knowing α.

Same result for fixed $f_t = f$ (classical optimisation) even without curvature via **derivative condition**.

Reason

Curvature implies $\Omega(V_T)$ cumulative slack between loss and its tangent lower bound.

MetaGrad Adapts to Stochastic Margin



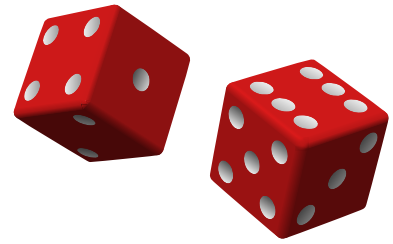
Consider i.i.d. losses $f_t \sim \mathbb{P}$ with **stochastic optimum**

$$\boldsymbol{u}^* \;=\; \arg\min_{\boldsymbol{u}} \mathbb{E}\, f(\boldsymbol{u})$$

Goal is small **pseudo-regret** compared to $\boldsymbol{u}^*$:

$$R_T^* \;=\; \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{u}^*)$$

MetaGrad Adapts to Stochastic Margin



Consider i.i.d. losses $f_t \sim \mathbb{P}$ with **stochastic optimum**

$$\boldsymbol{u}^* \;=\; \arg\min_{\boldsymbol{u}} \mathbb{E}\, f(\boldsymbol{u})$$

Goal is small **pseudo-regret** compared to $\boldsymbol{u}^*$:

$$R_T^* \;=\; \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{u}^*)$$

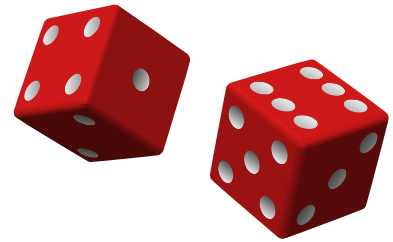
Corollary

*For any β-*__Bernstein__ $\mathbb{P}$*, MetaGrad keeps the expected regret below*

$$\mathbb{E}\, R_T^* \;\leq\; O\left((d \ln T)^{\frac{1}{2-\beta}}\; T^{\frac{1-\beta}{2-\beta}} \right).$$

Fast rates without curvature: e.g. absolute loss, hinge loss, ...

MetaGrad Adapts to Stochastic Margin



Consider i.i.d. losses $f_t \sim \mathbb{P}$ with **stochastic optimum**

$$\boldsymbol{u}^* \;=\; \arg\min_{\boldsymbol{u}} \mathbb{E}\, f(\boldsymbol{u})$$

Goal is small **pseudo-regret** compared to $\boldsymbol{u}^*$:

$$R_T^* \;=\; \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{u}^*)$$

Corollary

*For any β-***Bernstein** $\mathbb{P}$, *MetaGrad keeps the expected regret below*

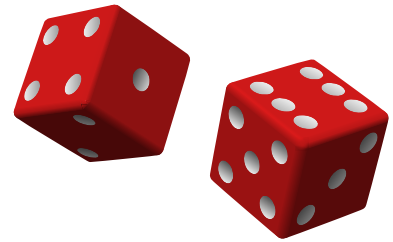
$$\mathbb{E}\, R_T^* \;\leq\; O\left((d \ln T)^{\frac{1}{2-\beta}}\; T^{\frac{1-\beta}{2-\beta}} \right).$$

Fast rates without curvature: e.g. absolute loss, hinge loss, ...

Reason

Bernstein bounds $\mathbb{E}[V_T^*]$ above by $\mathbb{E}[R_T^*]$. "Solve" regret bound.

MetaGrad Adapts to Stochastic Margin



Consider i.i.d. losses $f_t \sim \mathbb{P}$ with **stochastic optimum**

$$\boldsymbol{u}^* \;=\; \arg\min_{\boldsymbol{u}} \mathbb{E}\, f(\boldsymbol{u})$$

Goal is small **pseudo-regret** compared to $\boldsymbol{u}^*$:

$$R_T^* \;=\; \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{u}^*)$$

Corollary

*For any β-***Bernstein** $\mathbb{P}$, *MetaGrad keeps the ex gret below*

$$\mathbb{E}\, R_T^* \;\leq\; O\left((d \ln T)^{\frac{1}{2-\beta}}\right.$$

Joint work with P. Grünwald
Come see more at poster #76

Fast rates without curvature: e.g. abso loss, hinge loss, ...

Reason

Bernstein bounds $\mathbb{E}[V_T^*]$ above by $\mathbb{E}[R_T^*]$. "Solve" regret bound.

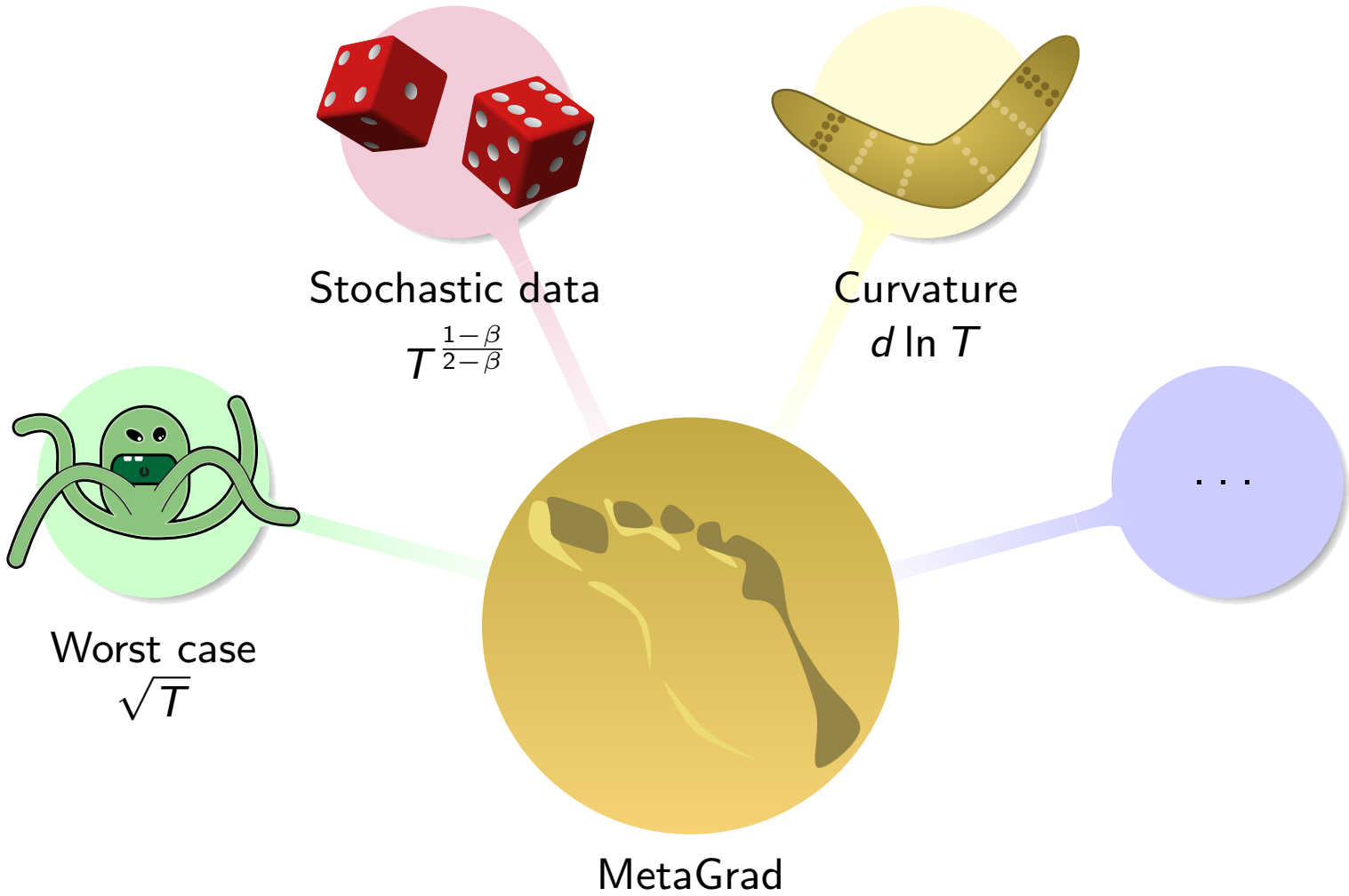
Conclusion

First contact with a new generation of **adaptive algorithms**.

Conclusion

First contact with a new generation of **adaptive algorithms**.

MetaGrad adapts to a wide range of environments:



Conclusion

First contact with a new generation of **adaptive algorithms**.

MetaGrad adapts to a wide range of ments: