Contributions to Deep Transfer Learning

From Supervised to Reinforcement Learning

Matthia Sabatelli
Montefiore Institute, Department of Electrical Engineering and Computer Science, Université de Liège, Belgium

March 30th 2022

Presentation outline

1. Part I: Preliminaries
   - Transfer Learning
2. Part II: Supervised Learning
   - On the Transferability of Convolutional Neural Networks
   - On the Transferability of Lottery Winners
3. Part III: Reinforcement Learning
   - A Novel Family of Deep Reinforcement Learning Algorithms
   - On the Transferability of Deep-Q Networks

PART I

Transfer Learning

Transfer Learning

On the one hand we have the concept of **source** S, which is needed to build some prior knowledge available for transfer

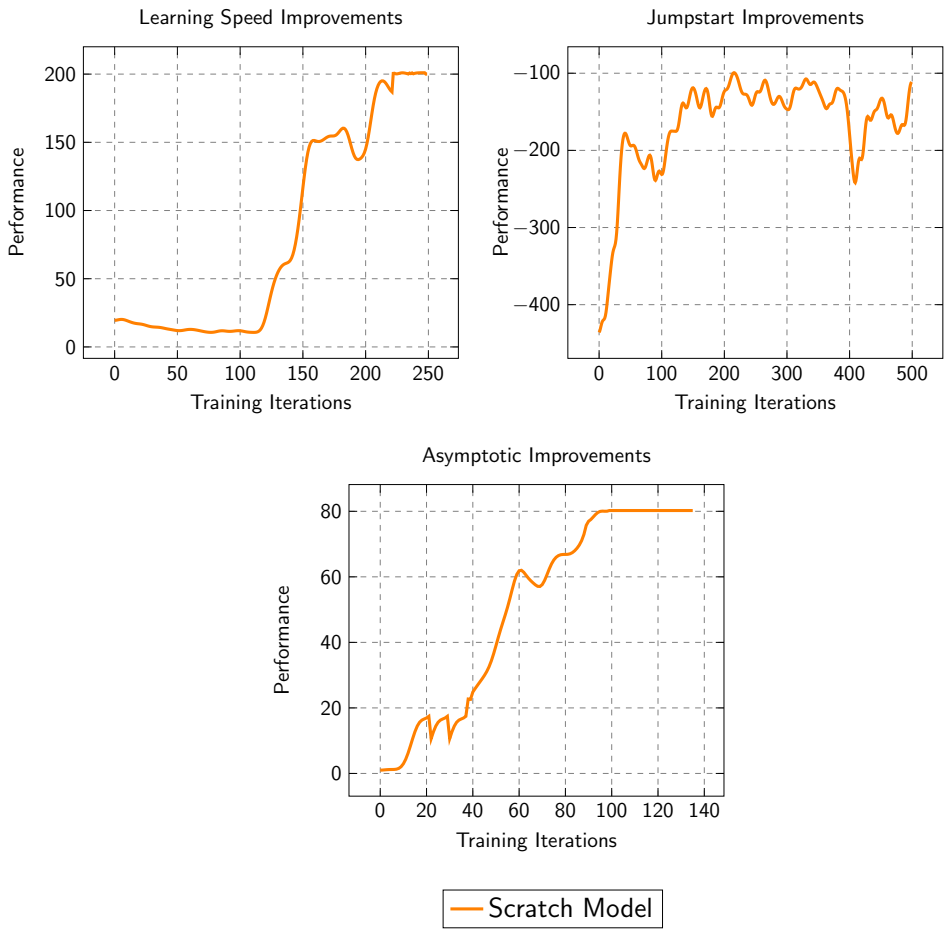
- Source Domain $\mathcal{D}_S$
- Source Task $\mathcal{T}_S$

While on the other hand we have the concept of **target** T, defining the machine learning problem we would like to solve

- Target Domain $\mathcal{D}_T$
- Target Task $\mathcal{T}_T$

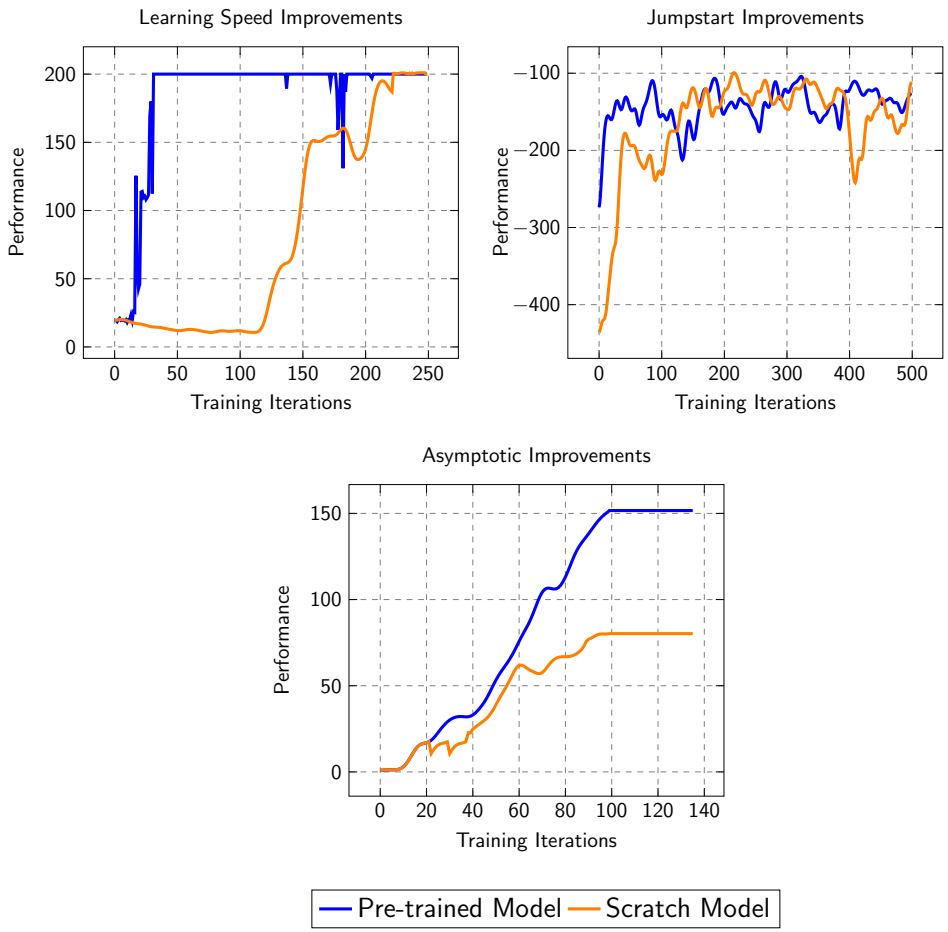
Transfer Learning

Transfer Learning in practice ...



Transfer Learning

Transfer Learning in practice ...



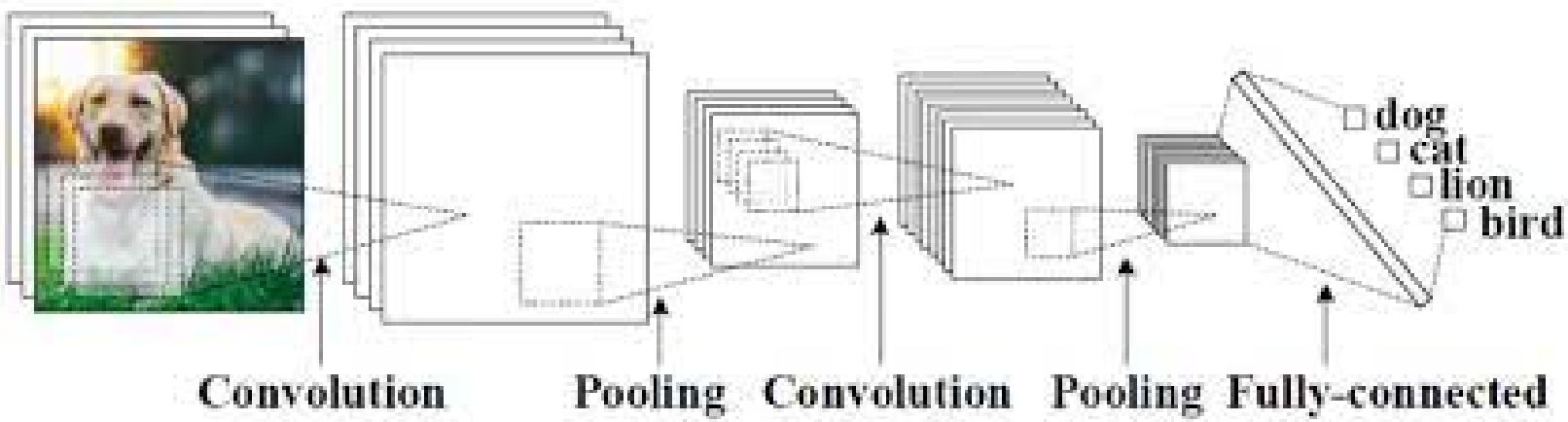
Transfer Learning

The family of machine learning models that is studied under the lens of transfer learning is that of Convolutional Neural Networks

- One of the main architectures when it comes to high-dimensional and spatially organized inputs
- Have achieved state-of-the-art results across many different machine learning problems
- We focus our analysis on Supervised Learning and Reinforcement Learning

Transfer Learning

When it comes to Supervised Learning ...



Transfer Learning

When it comes to Reinforcement Learning ...

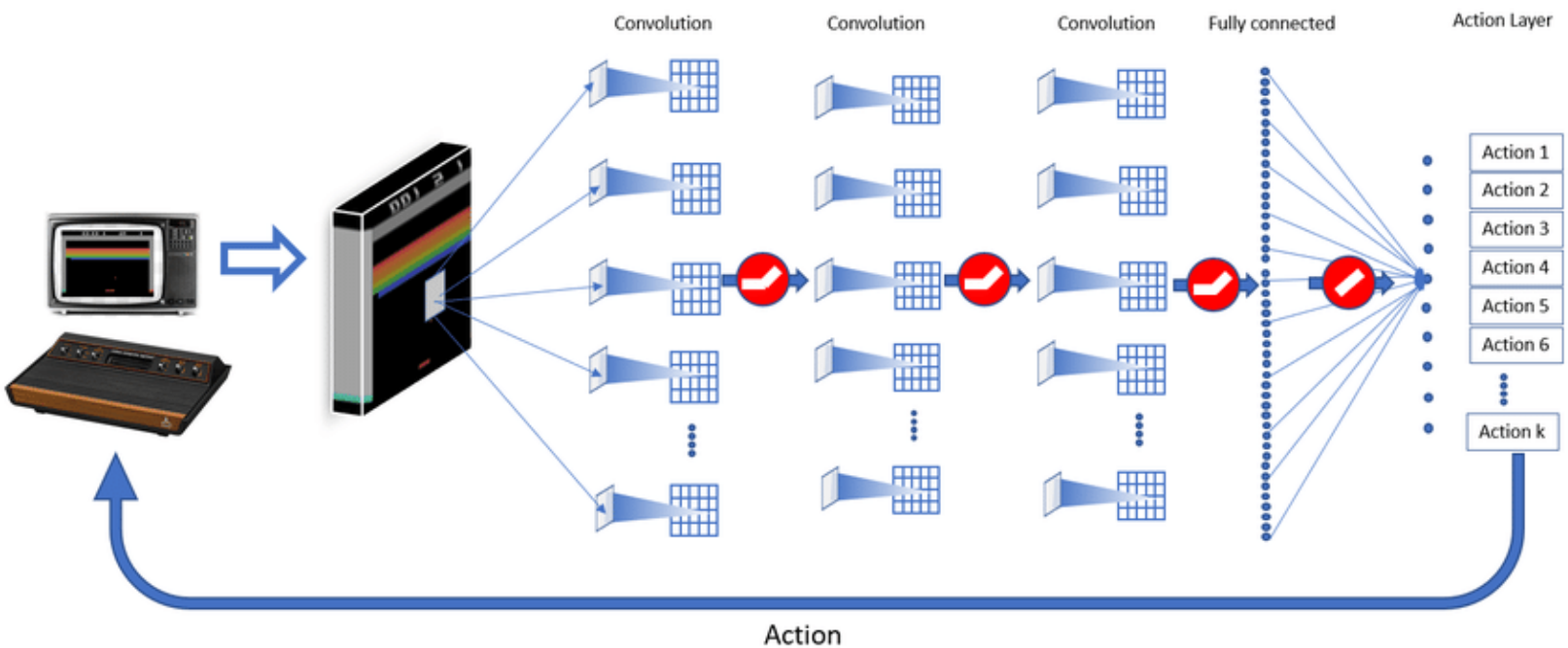


Figure: Image courtesy of Patel et al. (2019)

Transfer Learning

Why is Transfer Learning so interesting?

- It allows us to train deep learning models when there is a lack of training data
- Is useful when computational resources are scarce
- Results in better performing models
- Provides a unique tool for better understanding neural networks
- Having models that generalize and transfer across different domains and tasks is key for the development of Artificial Intelligence

PART II

We start by defining the components of Supervised Learning:

- An input space $\mathcal{X}$,
- An output space $\mathcal{Y}$,
- A joint probability distribution $P(X, Y)$
- A loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$

The goal is to find a function $f : \mathcal{X} \to \mathcal{Y}$ that minimizes the expected risk:

$$R(f) = \mathbb{E}_{(\mathrm{x},y)\sim P(X,Y)}\left[\ell(y, f(\mathrm{x}))\right]$$

On the Transferability of Convolutional Neural Networks

Typically, the only information available to build this function f is a learning sample of input-output pairs $LS_T = \{(x_i, y_i) | i = 1, \ldots, N_T\}$ drawn independently from $P_T(X, Y)$.

However, when it comes to Transfer Learning we have an additional dataset LS_S that can be used for finding a better f than when only LS_T is used for training.

On the Transferability of Convolutional Neural Networks

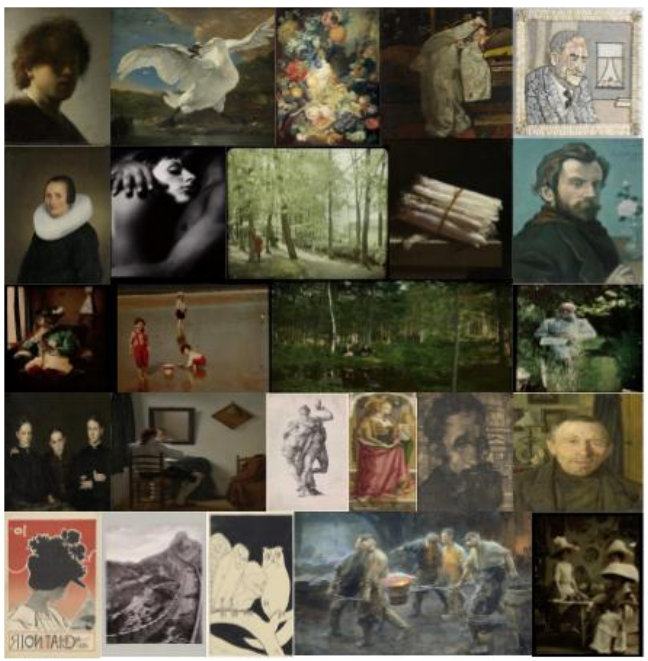
For our first study ...

- We consider f to come in the form of a Convolutional Neural Network (CNN)
- We define the source domain $\mathcal{D}_S$ to be that of natural images
- We define the target domain $\mathcal{D}_T$ to be that of Digital Heritage
- We assume that labels are available in both the source and target data and that the input spaces $\mathcal{X}_T$ and $\mathcal{X}_S$ match: Inductive Transfer Learning

On the Transferability of Convolutional Neural Networks

As source-task $\mathcal{T}_S$ we have a CNN pre-trained on the ImageNet dataset

We have three target-tasks $\mathcal{T}_T$ defined by the Rijksmuseum collection



On the Transferability of Convolutional Neural Networks

More specifically we aim to classify the artworks of the Rijksmuseum by:

Material: $\mathcal{T}_T$ ①

Type: $\mathcal{T}_T$ ②

Artist: $\mathcal{T}_T$ ③

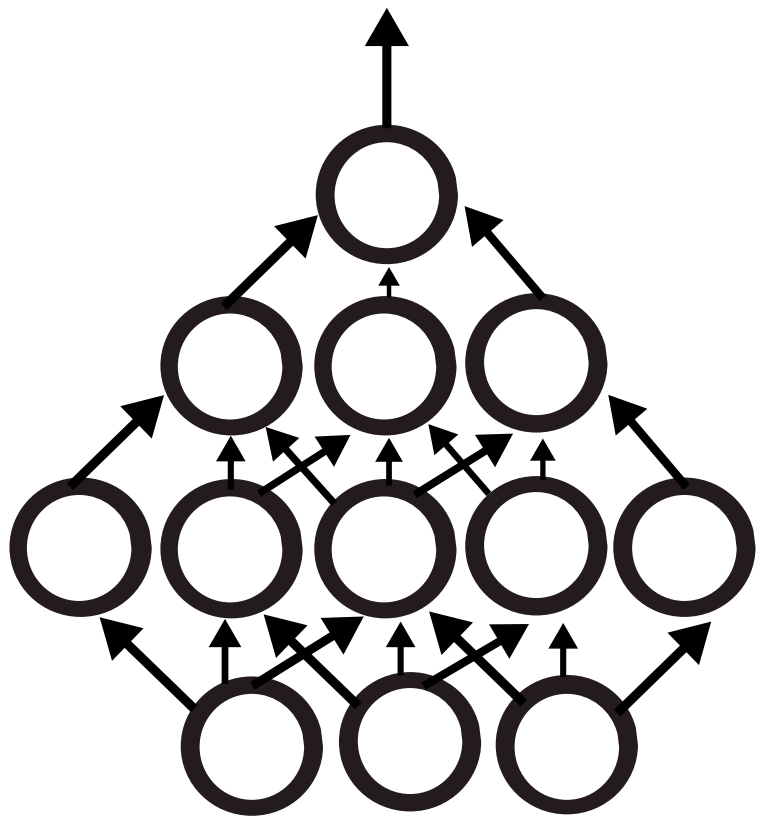
On the Transferability of Convolutional Neural Networks

$\Rightarrow$ Experimental setup ...

- We compare the transfer learning performance of four CNN architectures
  1. VGG19
  2. InceptionV3
  3. Xception
  4. ResNet50
- Explore three training strategies
  1. Random initialization (no transfer)
  2. Off-the-shelf feature extraction
  3. Fine-Tuning

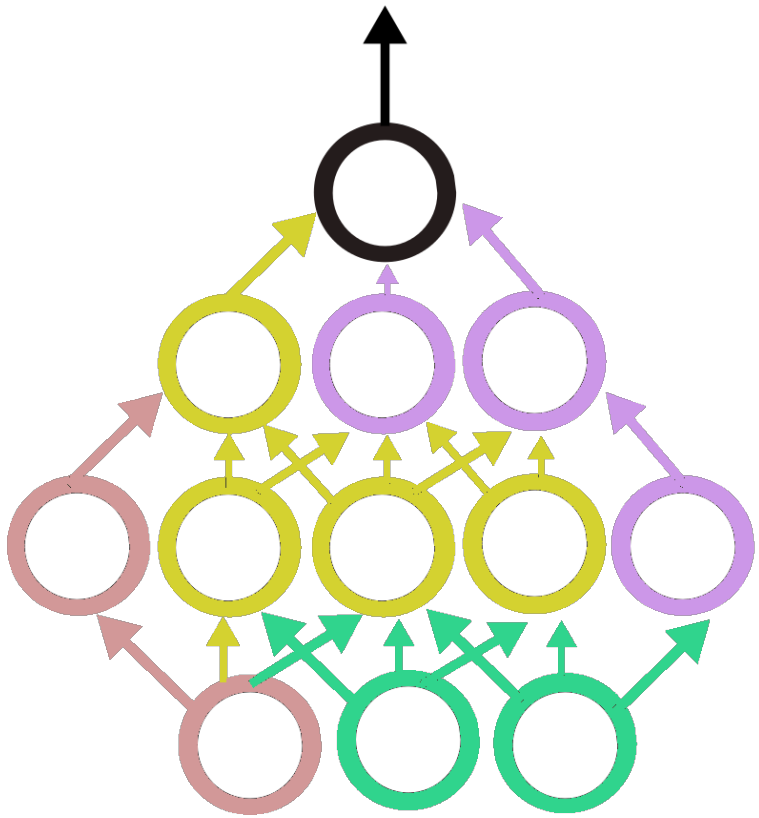
On the Transferability of Convolutional Neural Networks

- Let us consider a simplified neural network



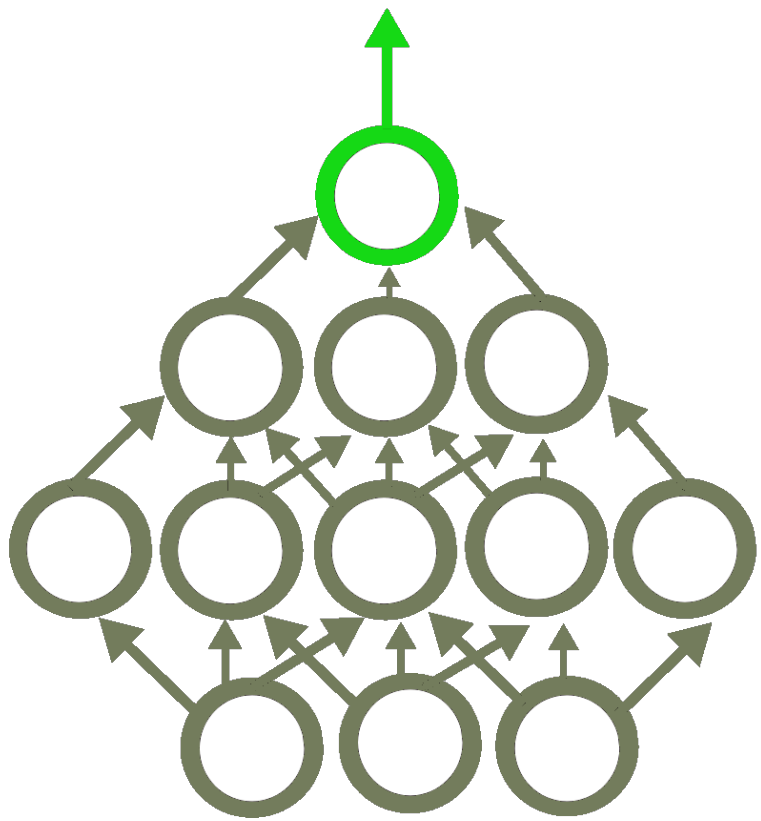
On the Transferability of Convolutional Neural Networks

- A randomly initialized model



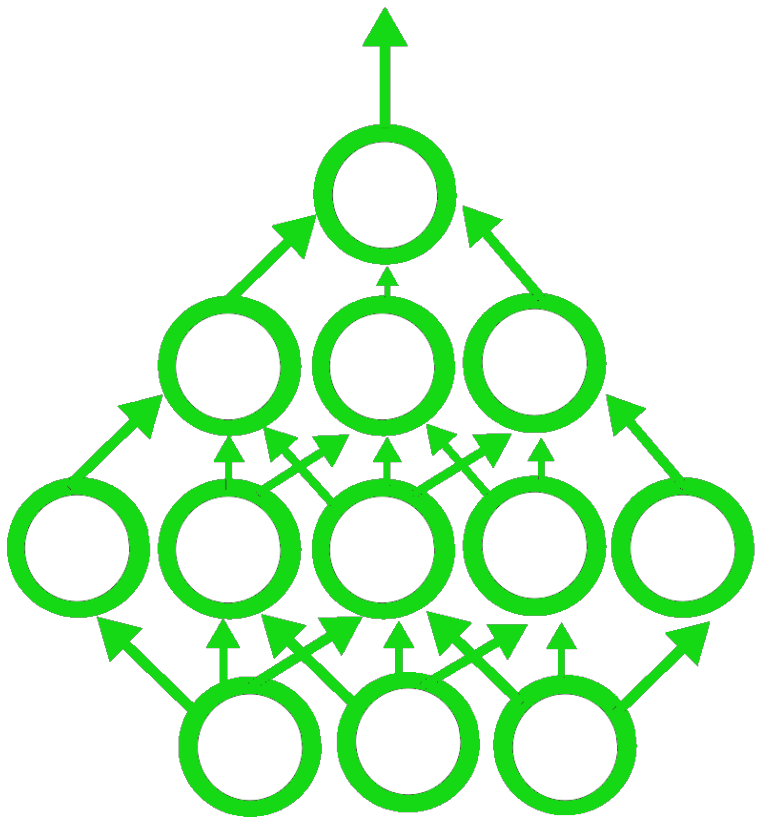
On the Transferability of Convolutional Neural Networks

- The off-the-shelf approach



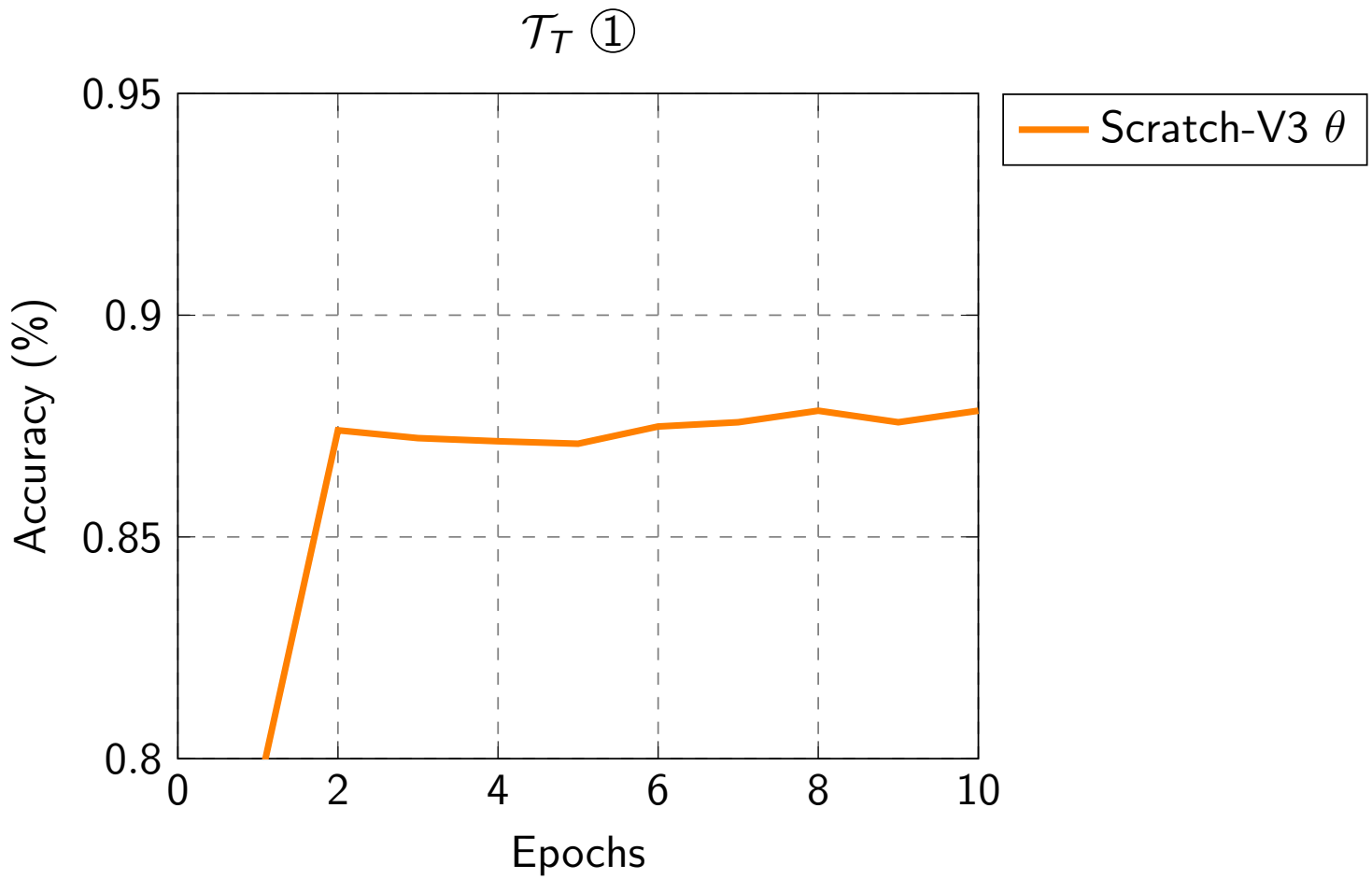
On the Transferability of Convolutional Neural Networks

- The fine-tuning approach



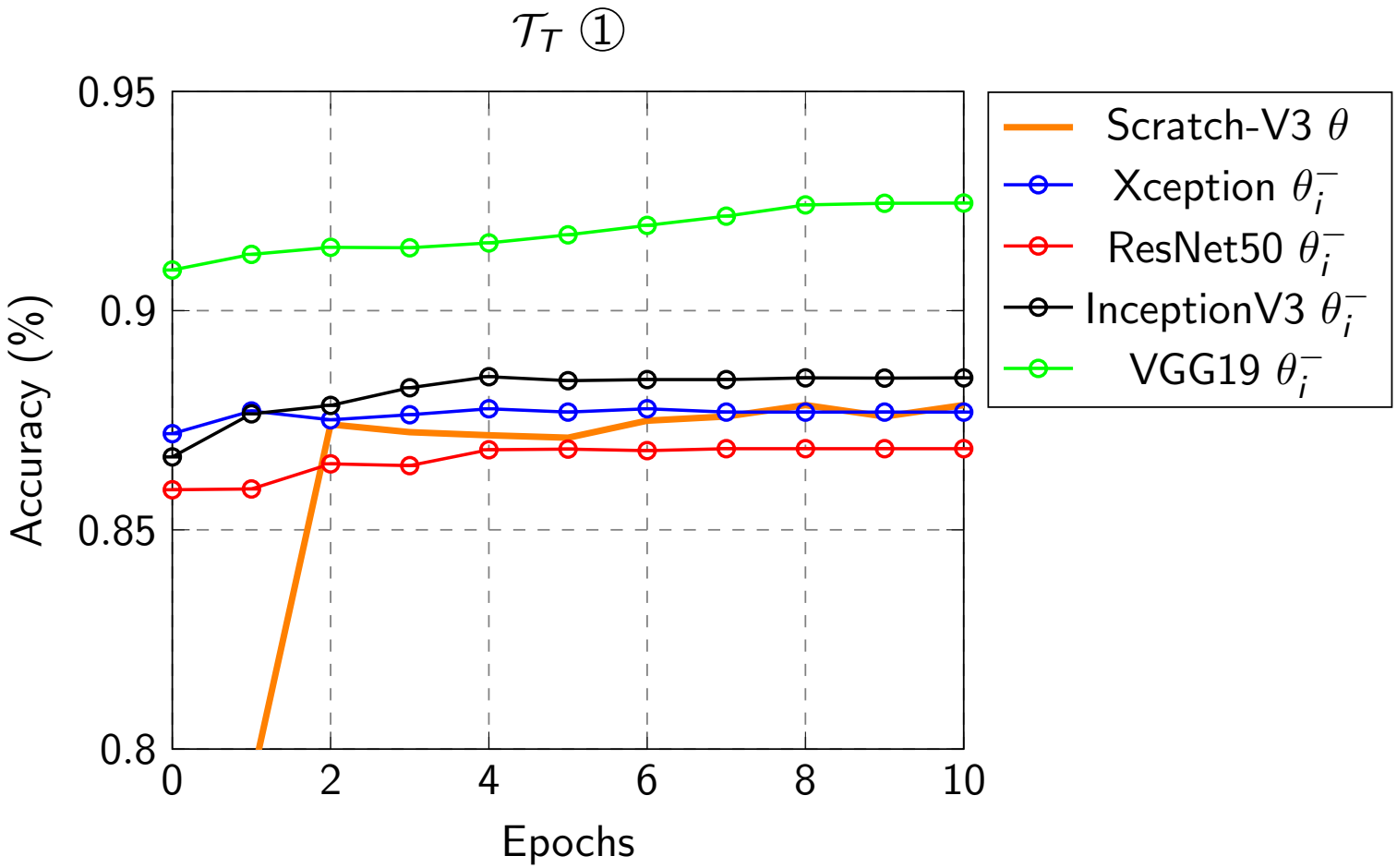
On the Transferability of Convolutional Neural Networks

The performance on $\mathcal{T}_T$ ①



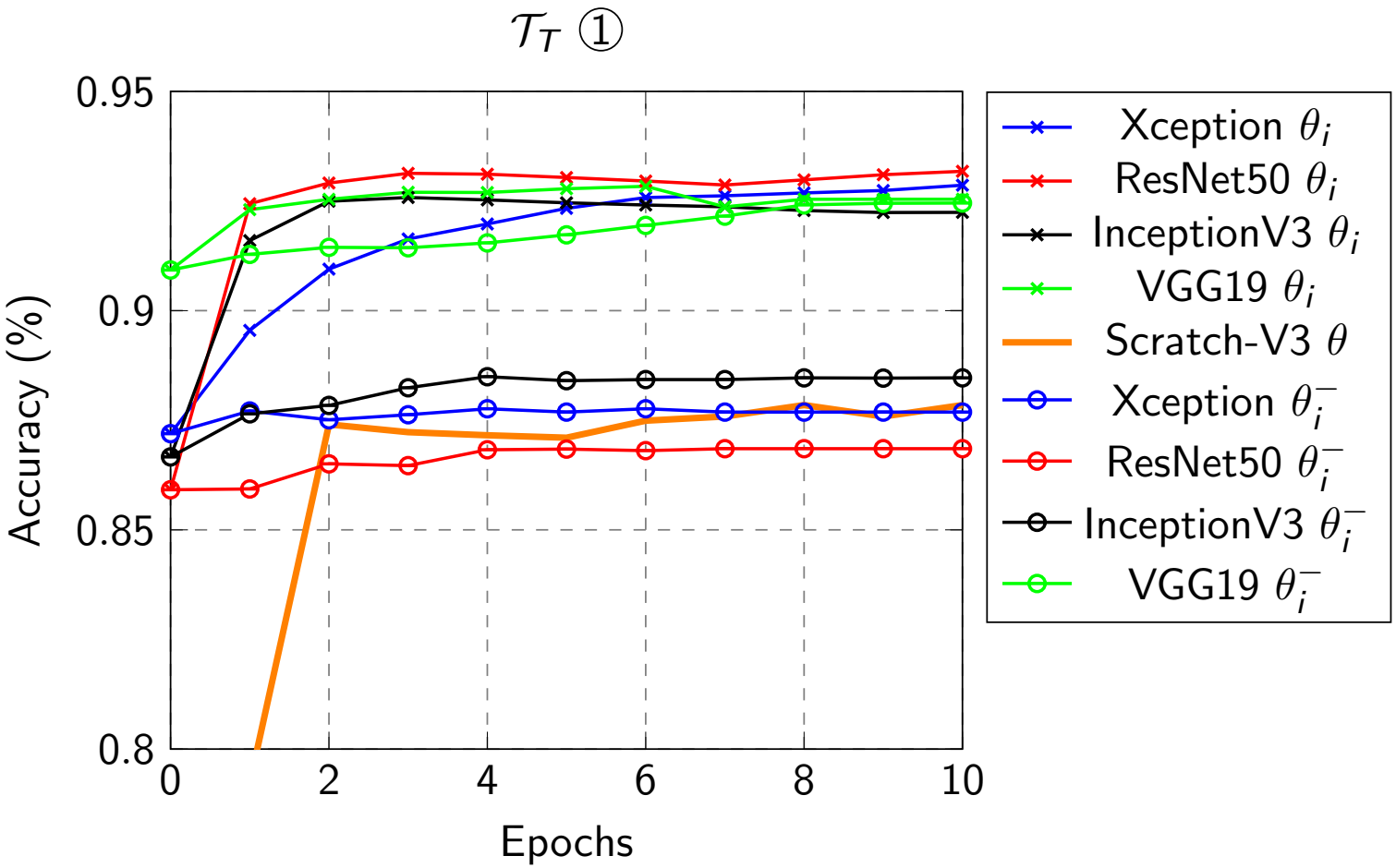
On the Transferability of Convolutional Neural Networks

The performance on $\mathcal{T}_T$ ①



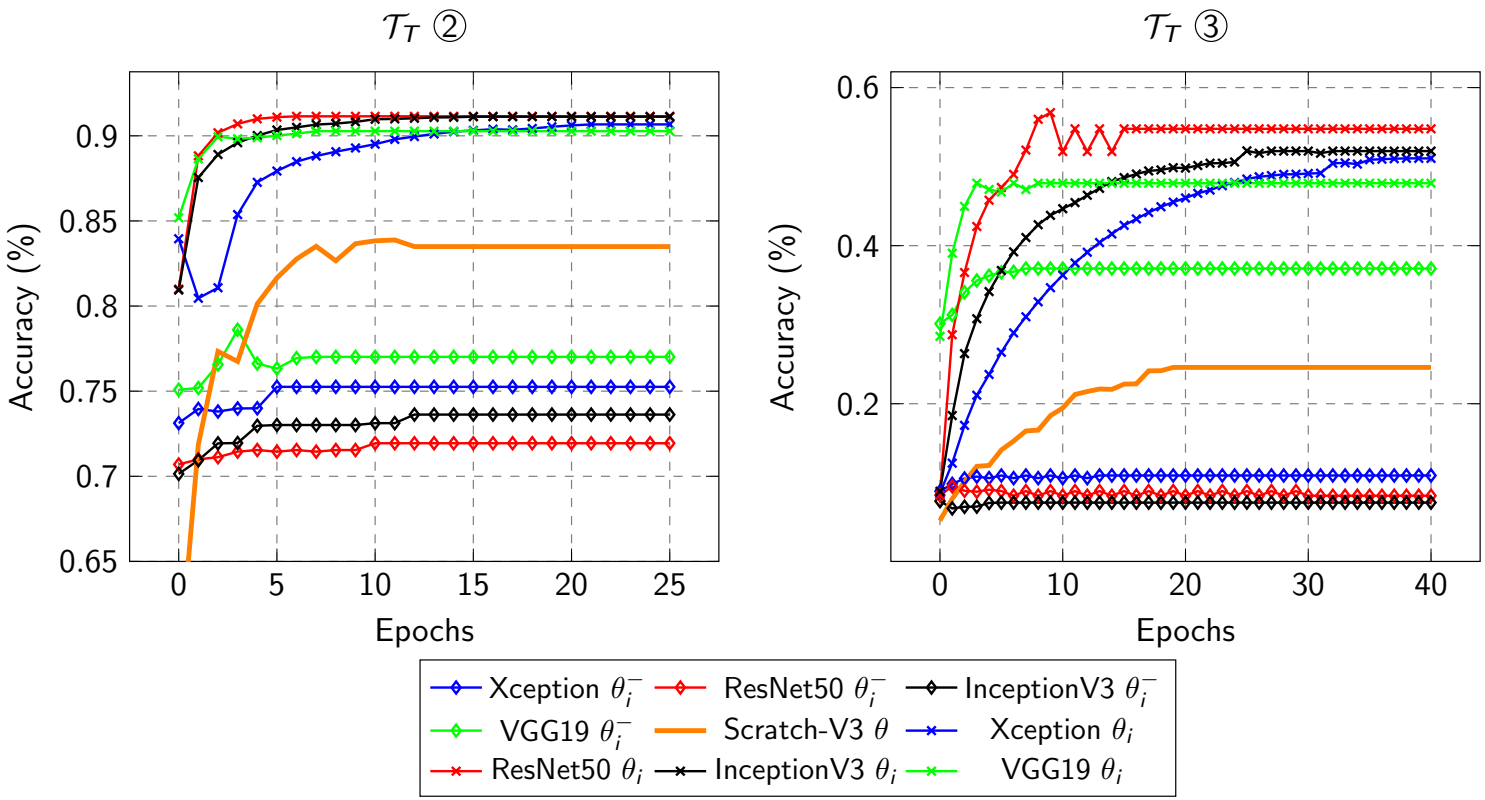
On the Transferability of Convolutional Neural Networks

The performance on $\mathcal{T}_T$ ①



On the Transferability of Convolutional Neural Networks

The performance on $\mathcal{T}_T$ ② and $\mathcal{T}_T$ ③



On the Transferability of Convolutional Neural Networks

We observe that:

- Both transfer learning approaches have significant benefits
- The off-the-shelf approach performs well for $\mathcal{T}_T$ ① and $\mathcal{T}_T$ ② but fails for $\mathcal{T}_T$ ③
- Fine-tuning the networks results in best performance

Next step ...

On the Transferability of Convolutional Neural Networks

$\Rightarrow$ After having fine-tuned the ImageNet models on the Rijksmuseum collection we investigate whether these models generalize to different, smaller, artistic collections

- We collected a new dataset of heritage objects present in the city of Antwerp
- Aim again at classifying their type $\mathcal{T}_T$ ②
- And the respective artist $\mathcal{T}_T$ ③
- Compare the performance to ImageNet pre-trained models only

On the Transferability of Convolutional Neural Networks

Table: The results obtained on the classification experiments performed on the Antwerp dataset with models that have been initially pre-trained on ImageNet (θ_i) and the same architectures which have been fine tuned on the Rijksmuseum dataset (θ_r). Our results show that the latter pre-trained networks yield better results both if used as off-the-shelf feature extractors and if fine-tuned.

$\mathcal{T}_T$	model	θ_i + off-the-shelf	θ_r + off-the-shelf	θ_i + fine-tuning	θ_r + fine-tuning
②	Xception	42.01%	62.92%	69.74%	72.03%
②	InceptionV3	43.90%	57.65%	70.58%	71.88%
②	ResNet50	41.59%	64.95%	76.50%	78.15%
②	VGG19	38.36%	60.10%	70.37%	71.21%
③	Xception	48.52%	54.81%	58.15%	58.47%
③	InceptionV3	21.29%	53.41%	56.68%	57.84%
③	ResNet50	22.39%	31.38%	62.57%	69.01%
③	VGG19	49.90%	53.52%	54.90%	60.01%

On the Transferability of Convolutional Neural Networks

To conclude we show that

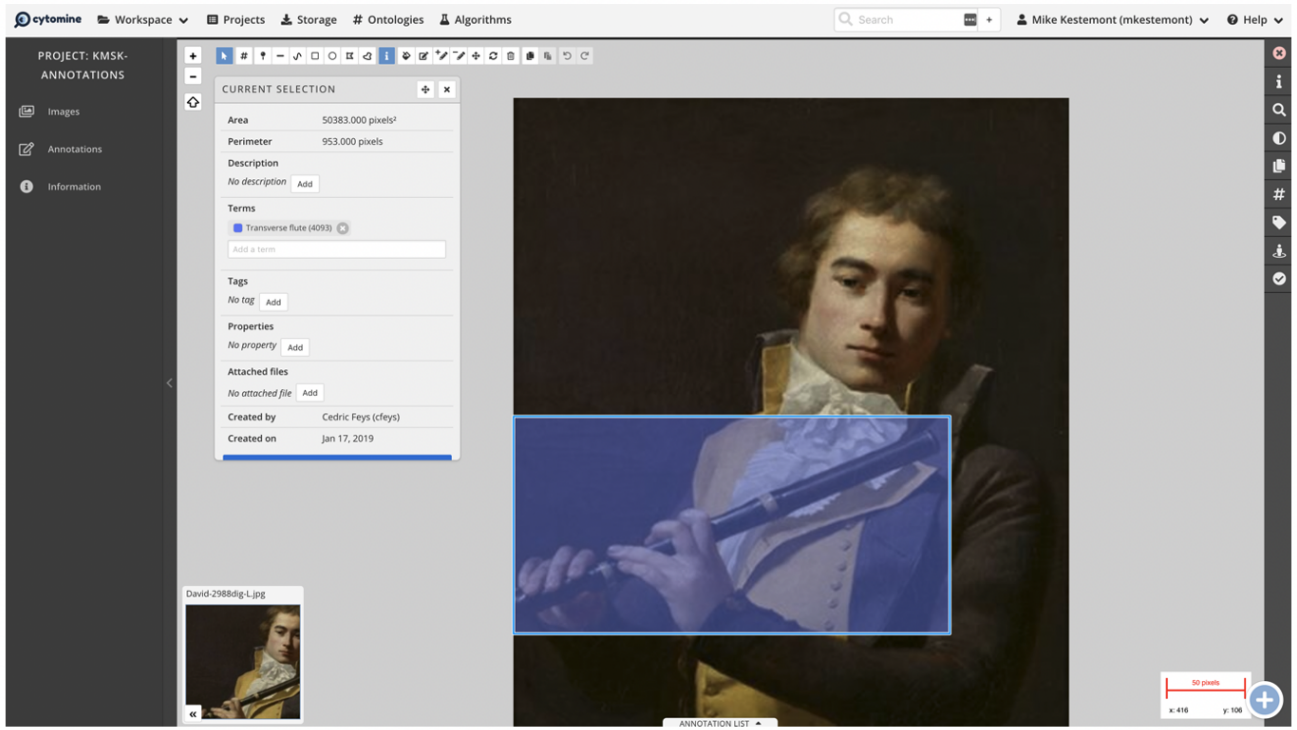
- Modern Convolutional Neural Networks exhibit strong transfer learning potential
- Even when the source task $\mathcal{T}_S$ (natural images) and the target task are very different $\mathcal{T}_T$ (art classification)
- Best performance is obtained after fine-tuning, which however comes at a computational cost

The present study only considers the computer vision task of image classification ...

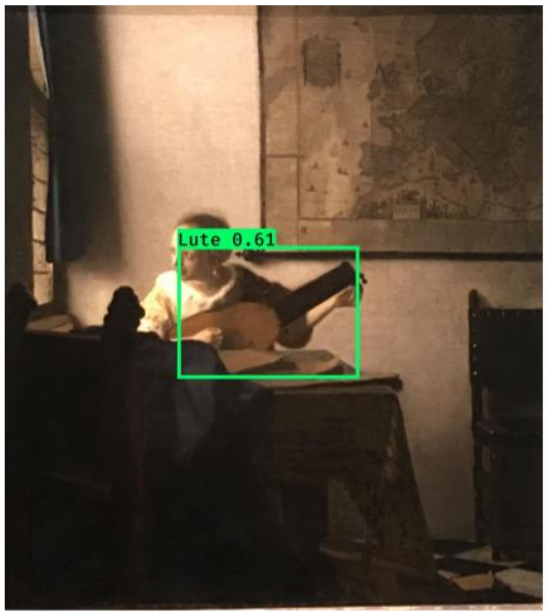
On the Transferability of Convolutional Neural Networks

$\Rightarrow$ Which is a limitation which we have addressed in our follow up work, where we perform a similar study for the computer vision task of object detection

For this purpose we have created the MINERVA dataset



On the Transferability of Convolutional Neural Networks



On the Transferability of Convolutional Neural Networks

In line with the classification experiments performed on the Rijksmuseum dataset, also when it comes to object detection we show that

- Pre-trained object detectors can get transferred across domains
- Fine-tuning these networks results in best performance
- This is the case for different families of object detection models ranging from YOLO, to networks that use region proposals and selective search [1].

[1] Claes, Yann. "Deep Learning for the Classification and Detection of Animals in Artworks." (2021).

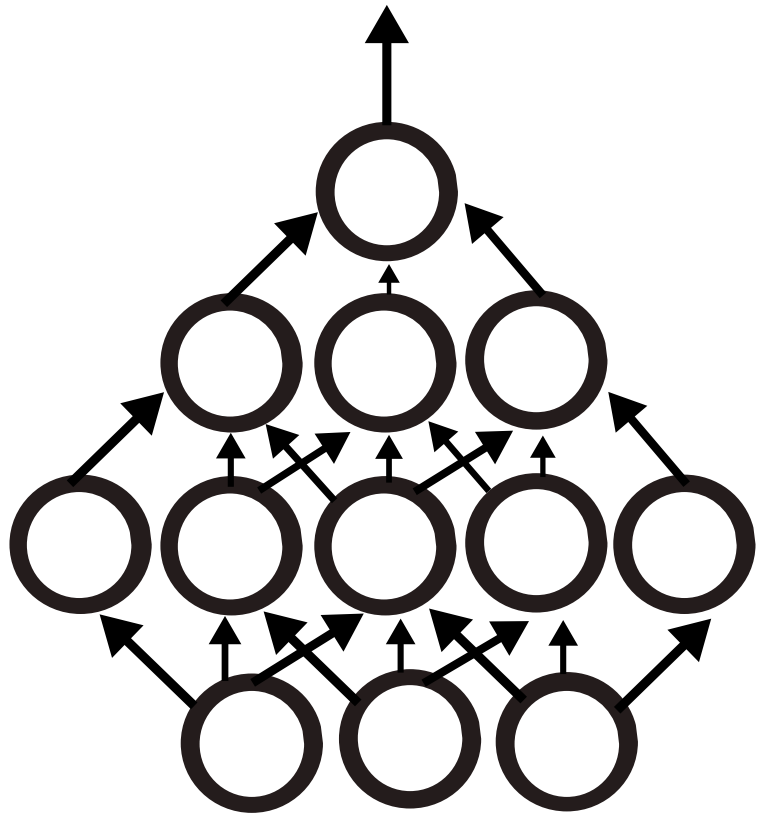
On the Transferability of Lottery Winners

On the Transferability of Lottery Winners

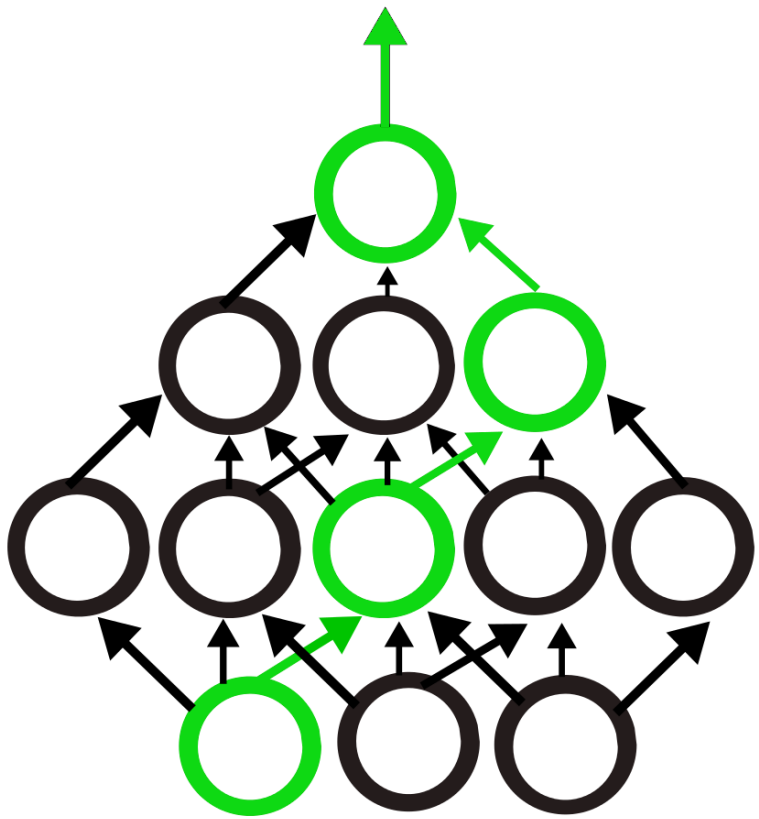
The Lottery Ticket Hypothesis: *"A randomly-initialized dense neural network contains a subnetwork that is initialized such that when trained in isolation it can match the test accuracy of the original network after training for at most the same number of iterations.*[2]*"*

[2] Frankle, Jonathan, and Michael Carbin. "The lottery ticket hypothesis: Finding sparse, trainable neural networks." (2018)

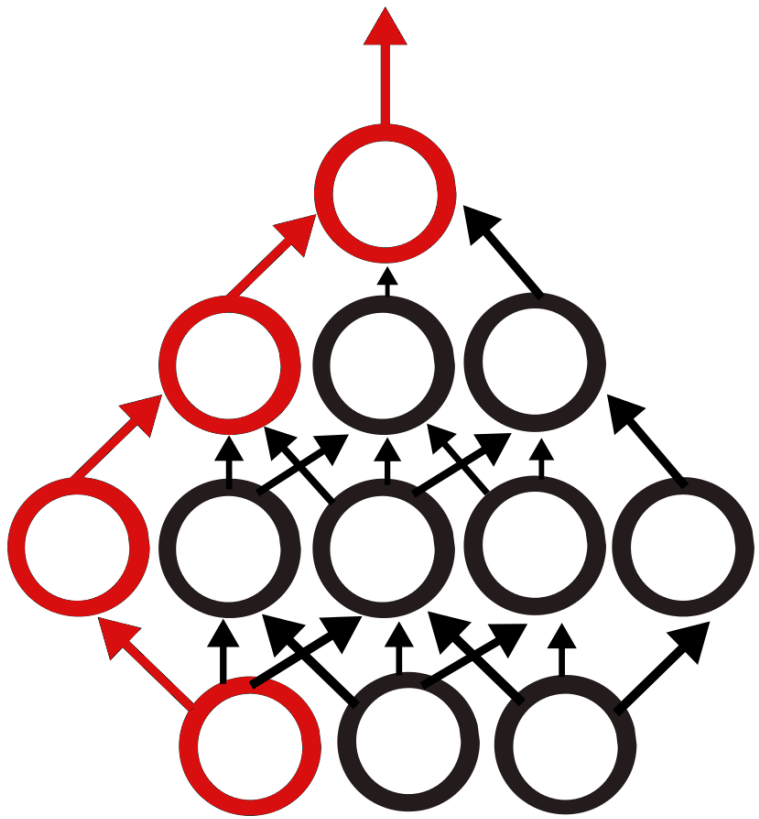
On the Transferability of Lottery Winners



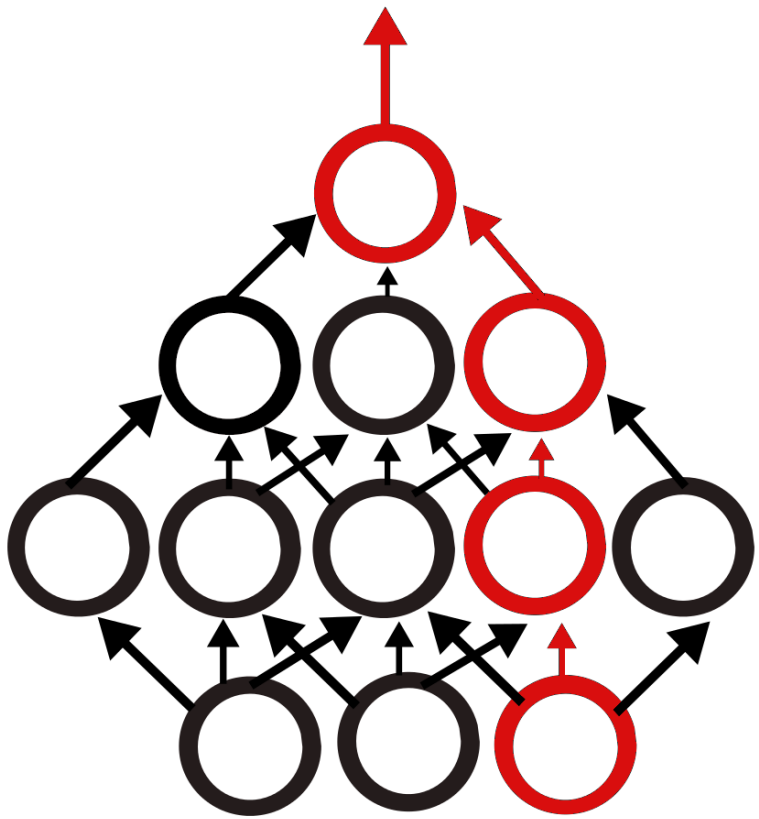
On the Transferability of Lottery Winners



On the Transferability of Lottery Winners



On the Transferability of Lottery Winners



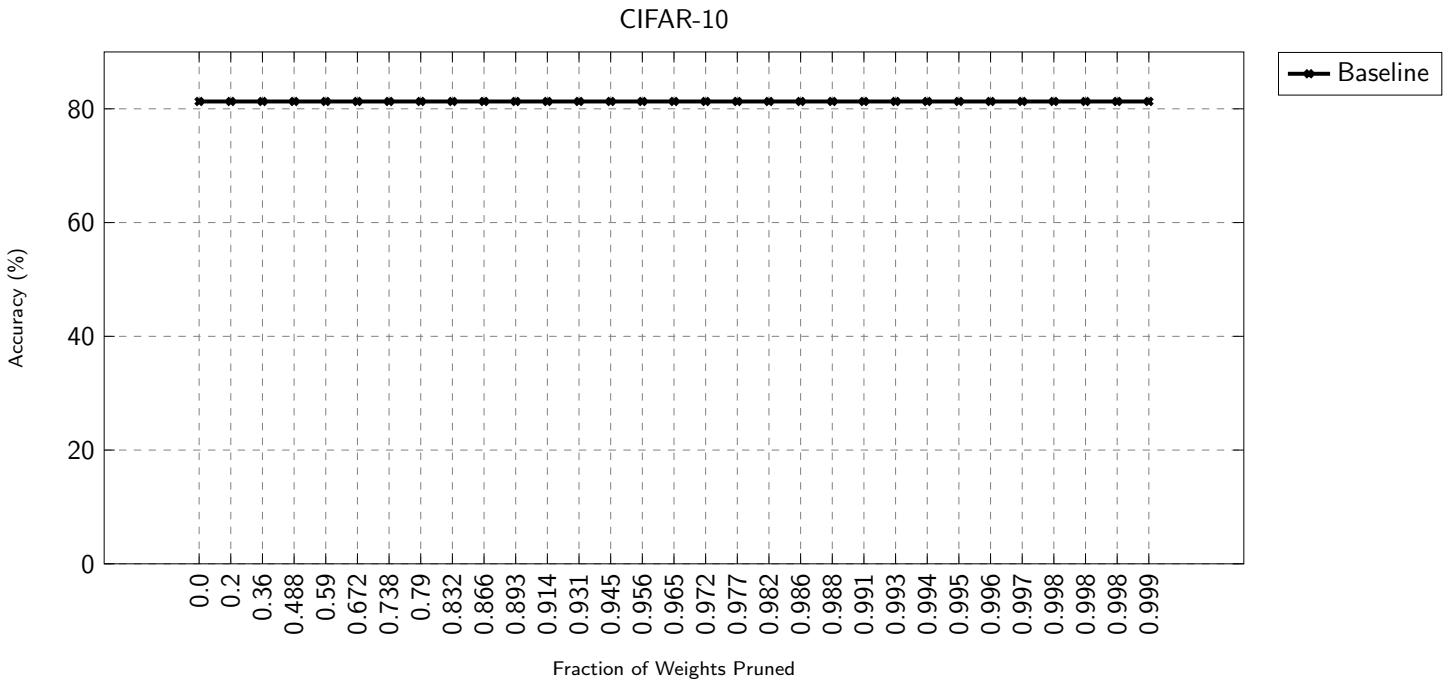
On the Transferability of Lottery Winners

$\Rightarrow$ How does one find a winning ticket?

- Randomly initialize a network $f(x; \theta_0)$ where $\theta_0 \sim \mathcal{D}_\theta$
- We train the network for j iterations
- We prune $p\%$ of the parameters in θ_j creating a mask m
- Reset the remaining parameters to their values at θ_0, creating a winning ticket $f(x; m \odot \theta_0)$

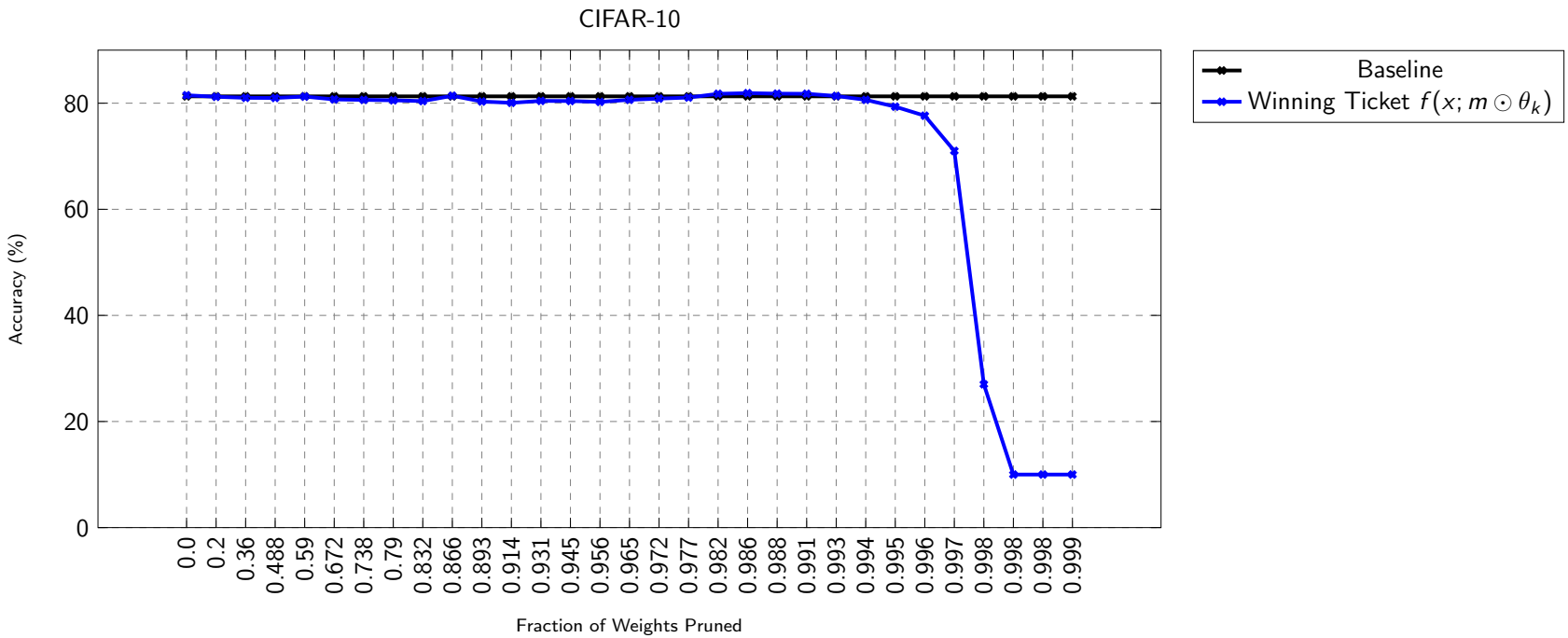
On the Transferability of Lottery Winners

The Lottery Ticket Hypothesis in practice ...



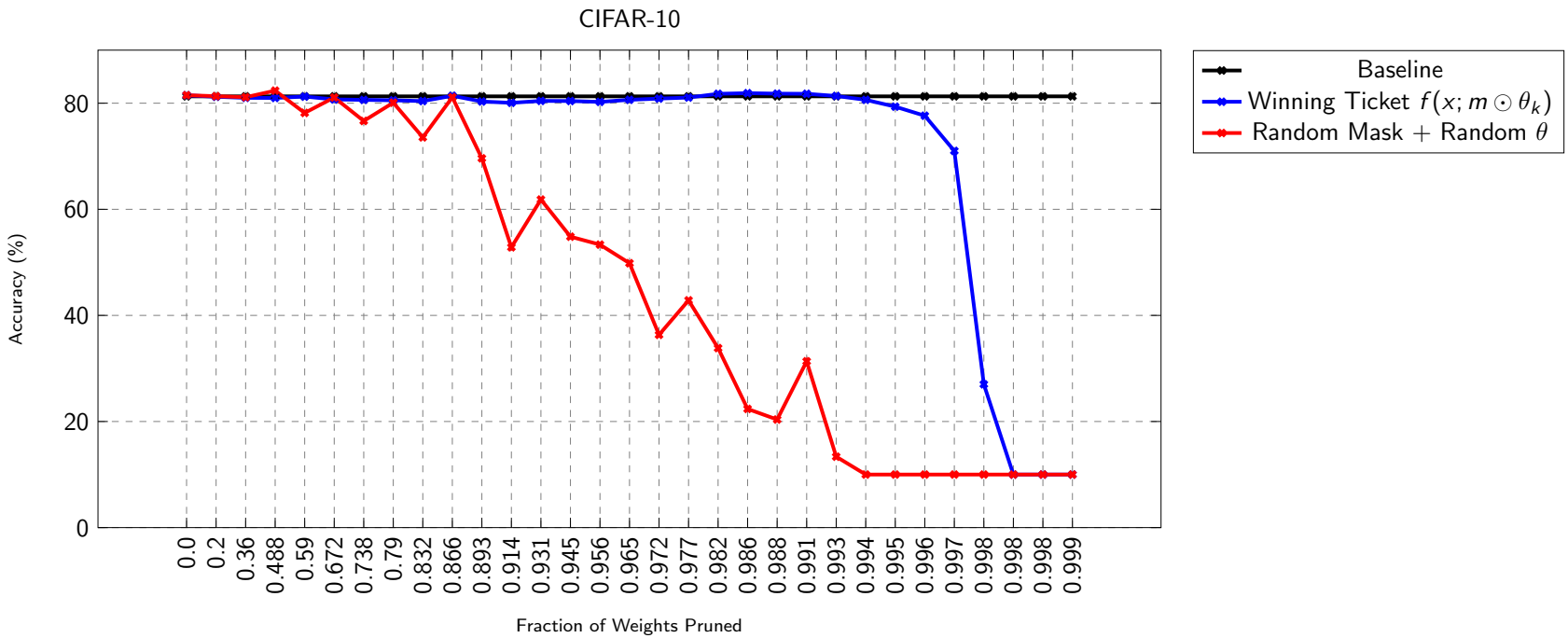
On the Transferability of Lottery Winners

The Lottery Ticket Hypothesis in practice ...



On the Transferability of Lottery Winners

The Lottery Ticket Hypothesis in practice ...



On the Transferability of Lottery Winners

Why are lottery tickets $f(x; m \odot \theta_0)$ so special?

- Train faster
- Faster Inference
- (Sometimes) obtain a better final performance

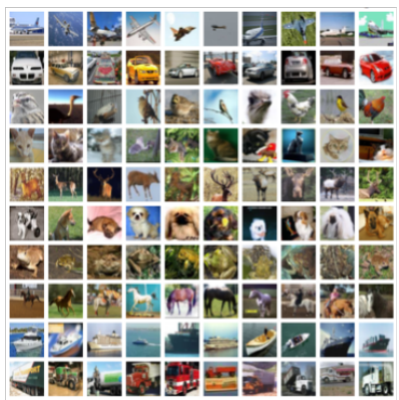
However ...

- Identifying a winning ticket is computationally expensive
- We do not know how and why lottery winners appear throughout learning

On the Transferability of Lottery Winners

$\Rightarrow$ Therefore we study whether winning tickets $f(x; m \odot \theta_0)$ found on natural image datasets can get transferred to the non-natural realm

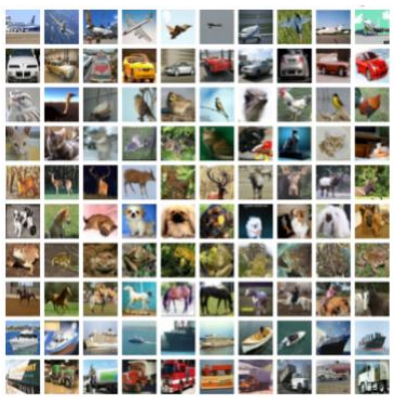
We use three popular Computer Vision datasets as source domains $\mathcal{D}_S$:



On the Transferability of Lottery Winners

$\Rightarrow$ Therefore we study whether winning tickets $f(x; m \odot \theta_0)$ found on natural image datasets can get transferred to the non-natural realm

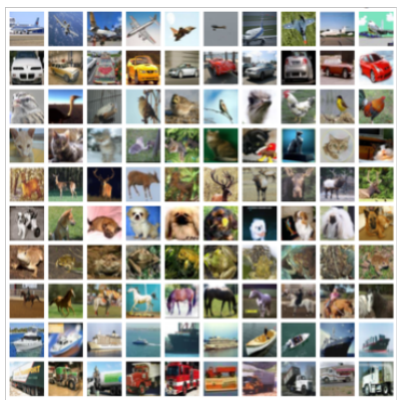
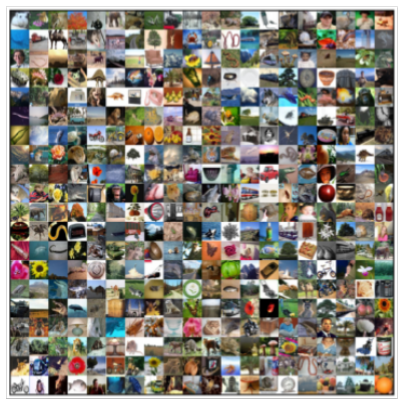
We use three popular Computer Vision datasets as source domains $\mathcal{D}_S$:



On the Transferability of Lottery Winners

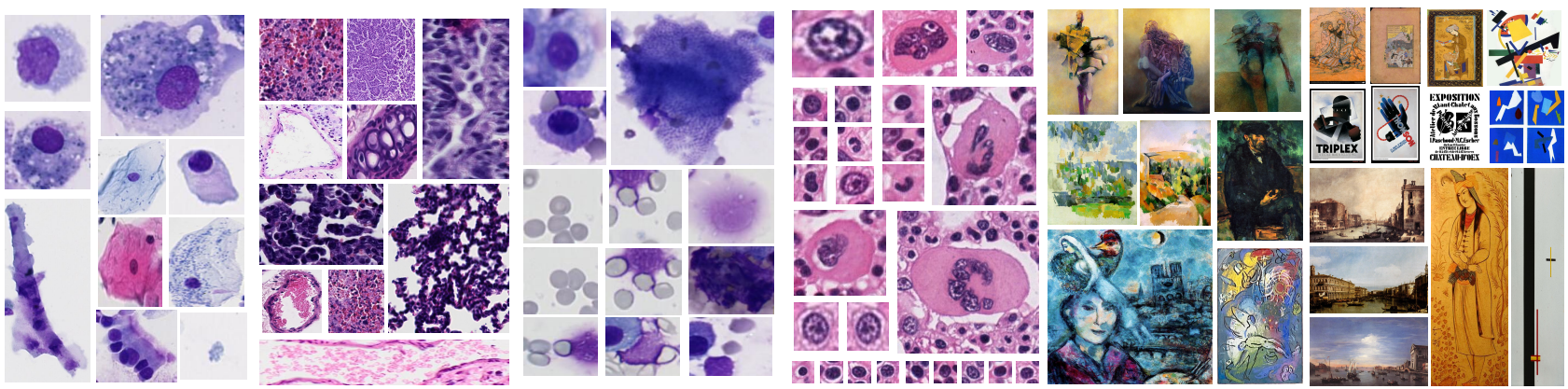
$\Rightarrow$ Therefore we study whether winning tickets $f(x; m \odot \theta_0)$ found on natural image datasets can get transferred to the non-natural realm

We use three popular Computer Vision datasets as source tasks $\mathcal{T}_S$

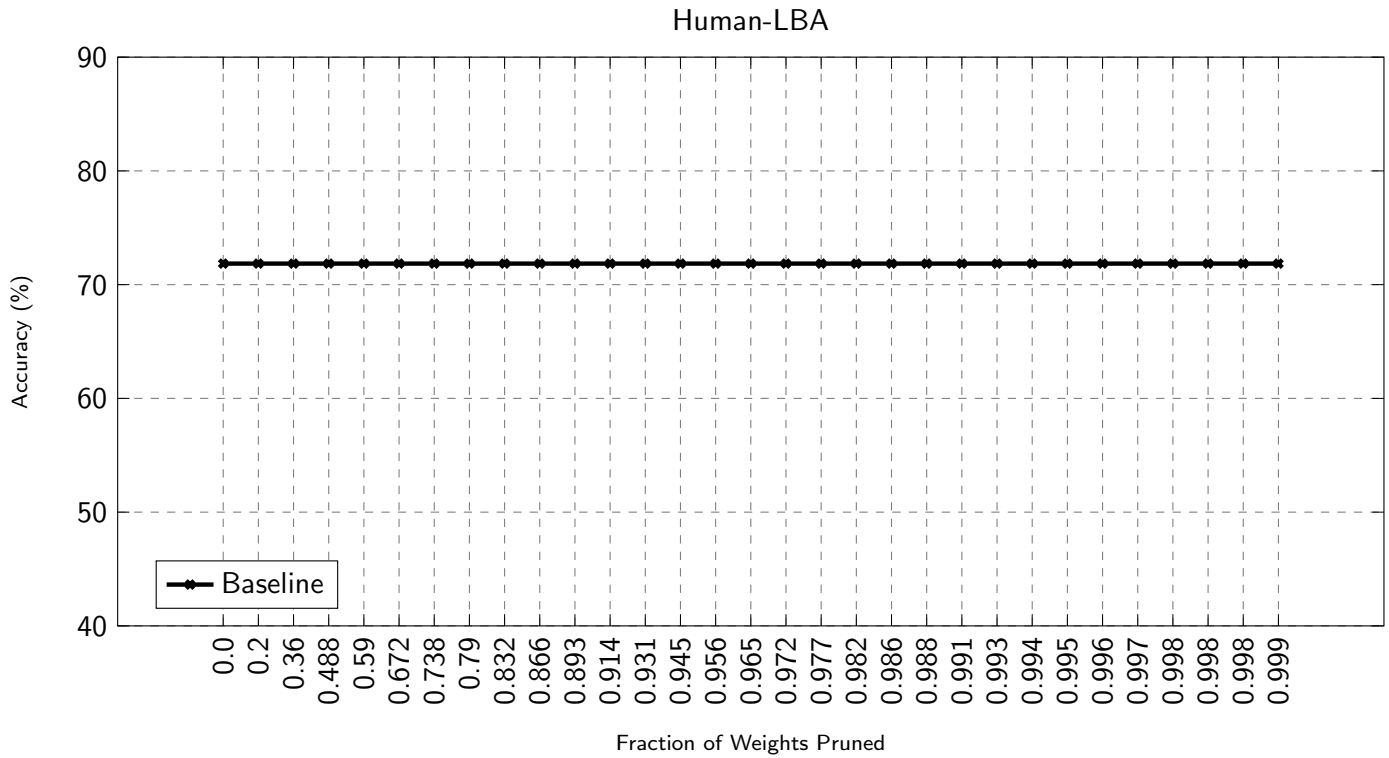

On the Transferability of Lottery Winners

And we use seven datasets of non-natural images as target tasks $\mathcal{T}_T$ coming from the fields of Digital Pathology and Digital Heritage



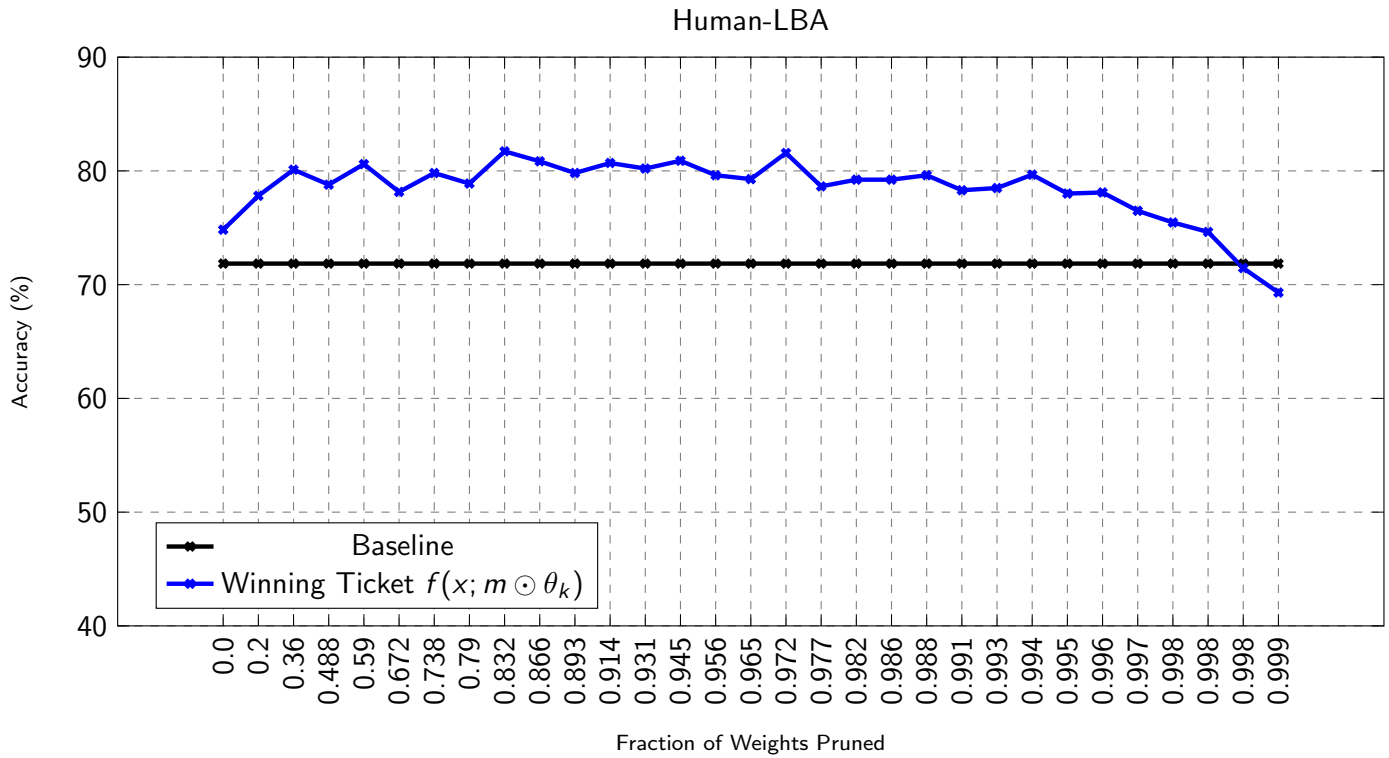
On the Transferability of Lottery Winners

Transferring lottery winners to the Digital Pathology data ...



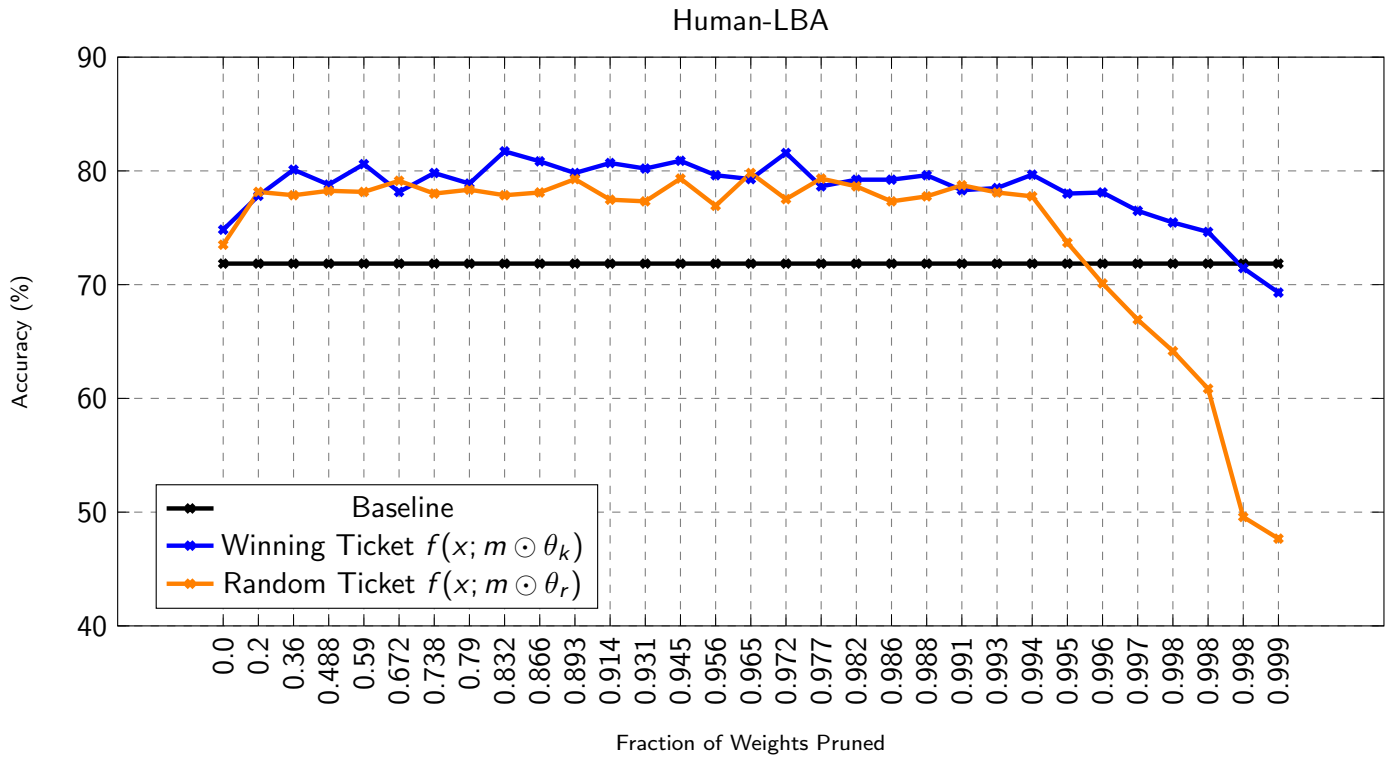
On the Transferability of Lottery Winners

Transferring lottery winners to the Digital Pathology data ...



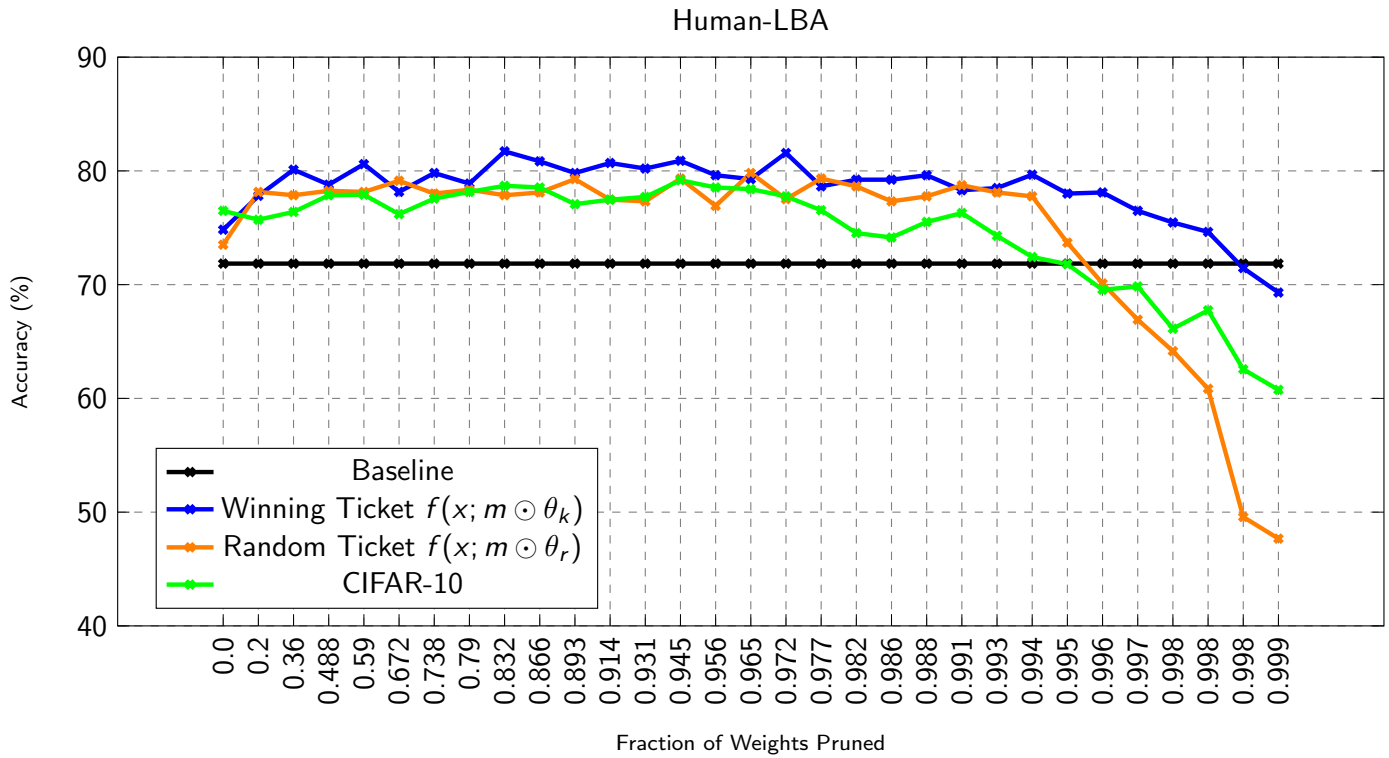
On the Transferability of Lottery Winners

Transferring lottery winners to the Digital Pathology data ...



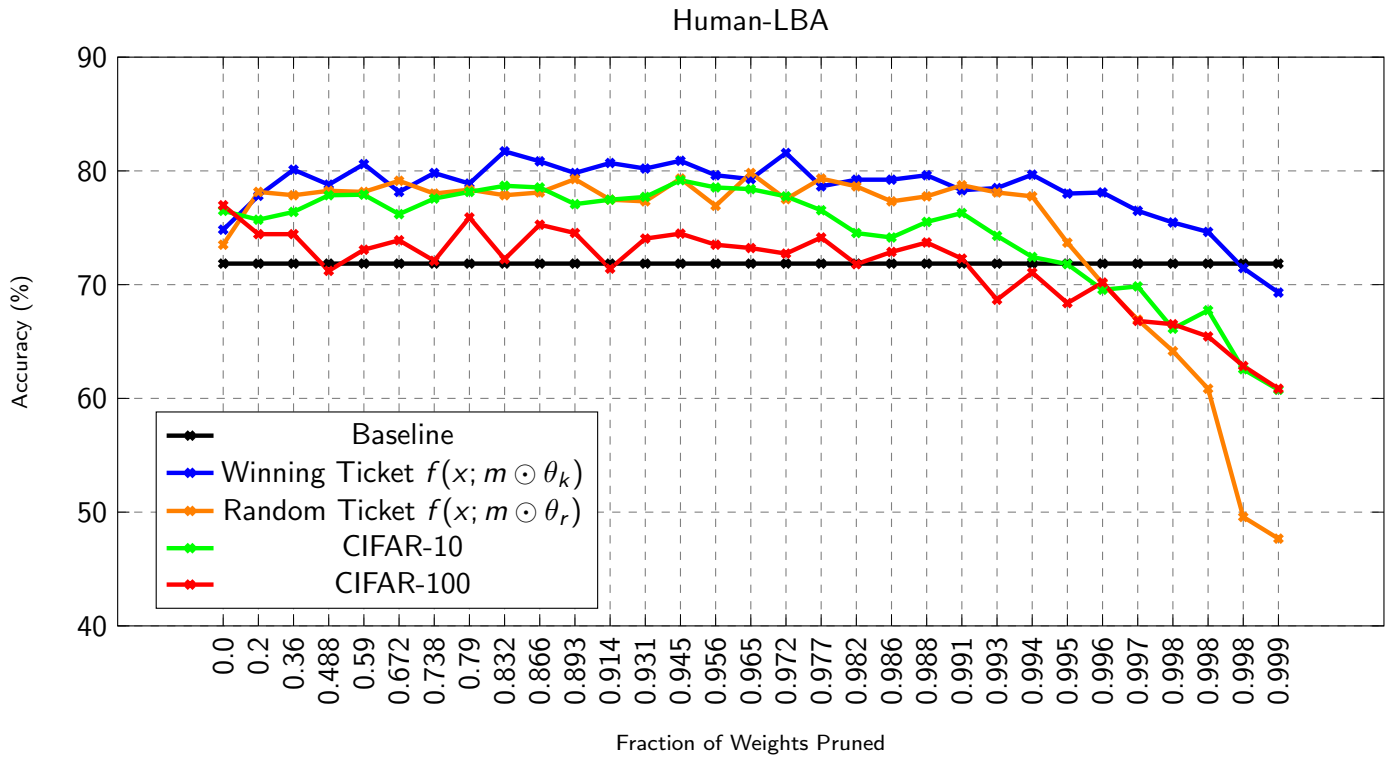
On the Transferability of Lottery Winners

Transferring lottery winners to the Digital Pathology data ...



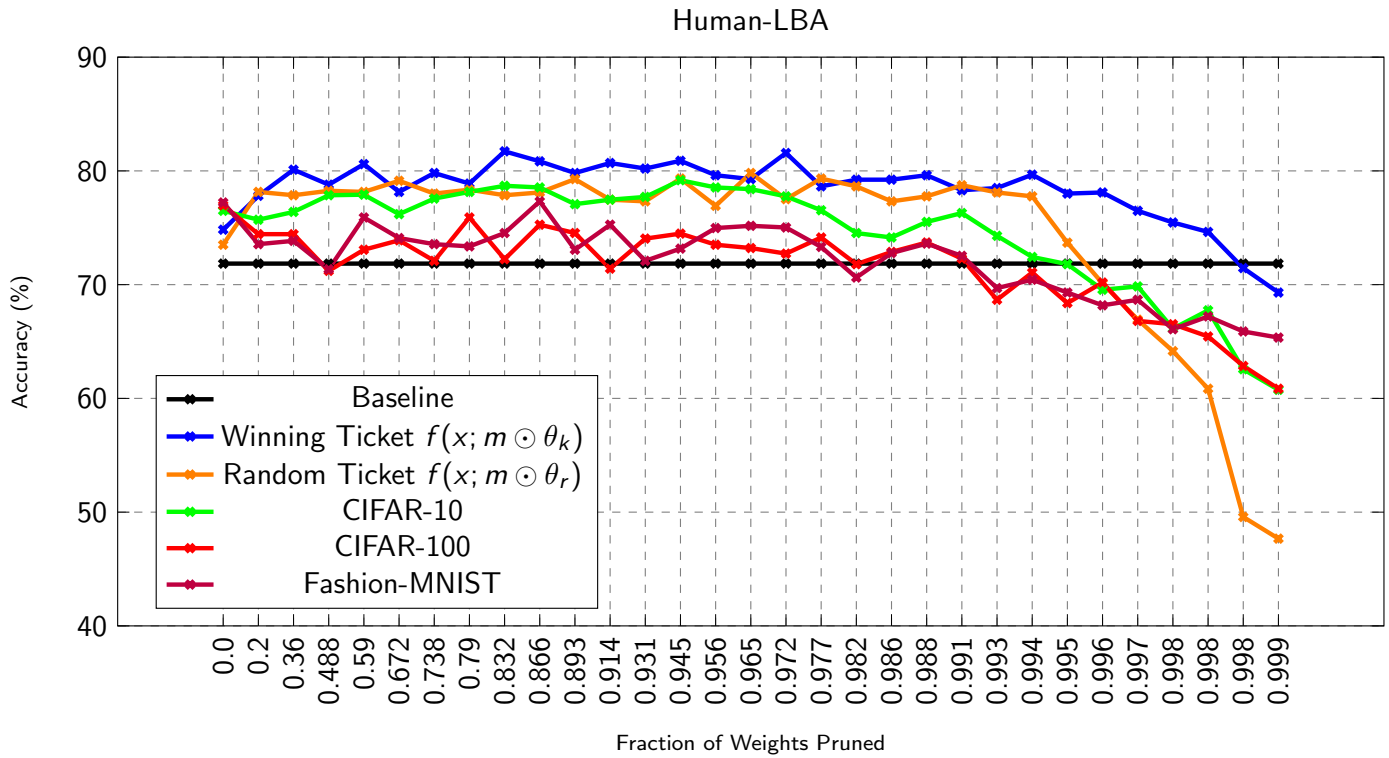
On the Transferability of Lottery Winners

Transferring lottery winners to the Digital Pathology data ...



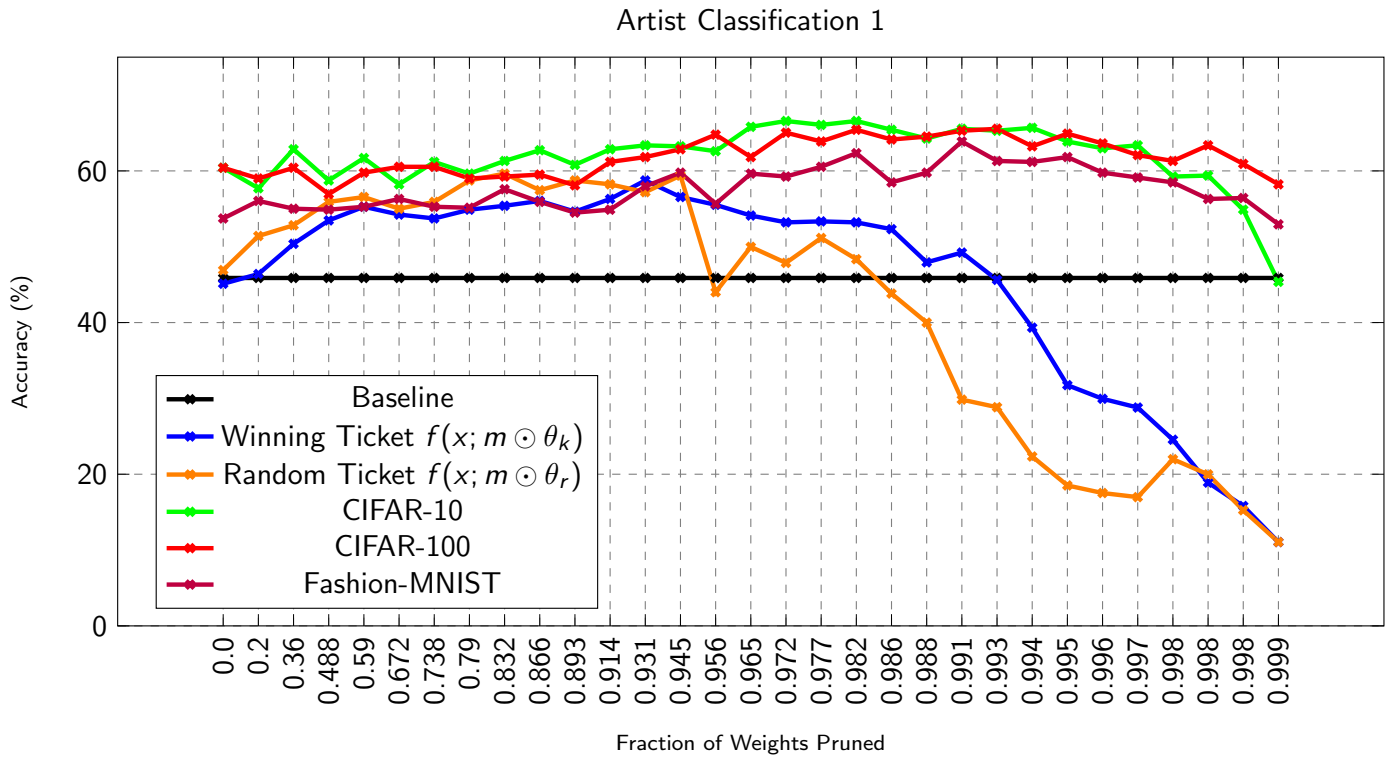
On the Transferability of Lottery Winners

Transferring lottery winners to the Digital Pathology data ...



On the Transferability of Lottery Winners

Transferring lottery winners to the Digital Heritage data ...



On the Transferability of Lottery Winners

Our main findings show that on Digital Pathology data

- All lottery winners significantly outperfom unpruned models
- Natural lottery winners contain a generic inductive bias (to some extent)
- Best performance is obtained by identifying a winning ticket directly on the target task $\mathcal{T}_T$

whereas on Digital Heritage data

- Natural lottery winners transfer much better
- They can even outperform target task $\mathcal{T}_T$ tickets

On the Transferability of Lottery Winners

We also provide additional empirical insights into the Lottery Ticket Hypothesis:

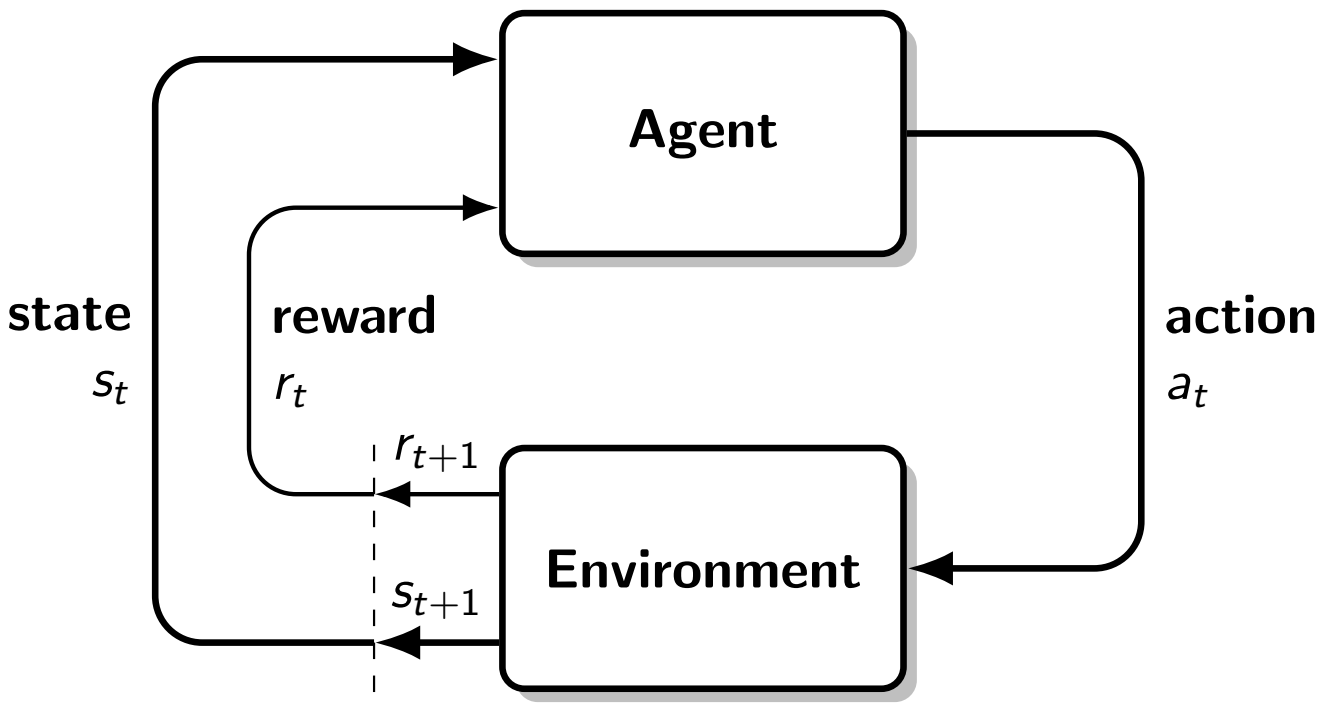
- We show that completely pre-trained winning tickets $f(x; m \odot \theta_i)$ overfit on the source task $\mathcal{T}_S$
- The presence of winning tickets does not depend on the size of the training data
- The closer the source task $\mathcal{S}_T$ and the target task $\mathcal{T}_T$ the better the transferability of $f(x; m \odot \theta_k)$

PART III

A Novel Family of Deep Reinforcement Learning Algorithms

A Novel Family of Deep Reinforcement Learning Algorithms

We now consider a different machine learning paradigm: Reinforcement Learning (RL), where an agent needs to learn how to interact with its environment



A Novel Family of Deep Reinforcement Learning Algorithms

Such interaction is modeled as a Markov Decision Process (MDP) consisting of the following elements:

- A set of possible states $\mathcal{S}$,
- A set of possible actions $\mathcal{A}$,
- A transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$,
- A reward function $\Re : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$,
- A discount factor denoted as $\gamma \in [0, 1]$.

Therefore we can define an MDP as $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \Re, \gamma \rangle$.

A Novel Family of Deep Reinforcement Learning Algorithms

The interaction between the agent and the environment is given by the agent's policy π, a probability distribution over $a \in \mathcal{A}(s)$ for each $s \in \mathcal{S}$:

$$\pi(a|s) = \Pr\{a_t = a | s_t = s\}, \text{ for all } s \in \mathcal{S} \text{ and } a \in \mathcal{A}.$$

The goal of the agent is to maximize the expected discounted return as:

$$\begin{aligned} G_t &= r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots \\ &= \sum_{k=0}^{\infty} \gamma^k r_{t+k}. \end{aligned}$$

A Novel Family of Deep Reinforcement Learning Algorithms

In RL some components of the MDP are unknown

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathfrak{R}, \gamma \rangle,$$

which in the case of model-free RL can be overcome by learning either the state-value function

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \middle| s_t = s, \pi \right],$$

or the state-action value function

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \middle| s_t = s, a_t = a, \pi \right].$$

A Novel Family of Deep Reinforcement Learning Algorithms

- Model-free algorithms are implemented in a tabular fashion, meaning that the state values, or state-action values, are stored within tables of sizes $|\mathcal{S}|$ and $|\mathcal{S} \times \mathcal{A}|$ respectively
- For many problems we typically seek to learn an approximation of the value functions

$$V^{\pi}(s) \approx V^{\pi}(s;\theta) \text{ and } Q^{\pi}(s,a;\theta) \approx Q(s,a;\theta)$$

This approximation can be represented by a Convolutional Neural Network

A Novel Family of Deep Reinforcement Learning Algorithms

Typical DRL algorithms aim at only learning an approximation of the state-action value function $Q^{\pi}(s, a) \approx Q^{\pi}(s, a; \theta)$

- A policy π can only be derived from $Q^{\pi}(s, a)$
- It is hard to combine RL algorithms with neural networks
- Dealing with one value function can be complicated enough

However ...

- Training can be very slow
- Algorithms are prone to diverge
- Do not correctly estimate $Q^{\pi}(s, a)$

A Novel Family of Deep Reinforcement Learning Algorithms

$\Rightarrow$ Therefore we suggest to jointly approximate the state-value function $V^{\pi}(s;\phi)$ alongside the state-action value function $Q^{\pi}(s,a;\theta)$

We can do this by either learning $V^{\pi}(s;\phi)$ with

$$L(\phi) = \mathbb{E}_{\langle s_t, a_t, r_t, s_{t+1}\rangle \sim U(D)}\left[\left(r_t + \gamma V(s_{t+1};\phi^{-}) - V(s_t;\phi)\right)^2\right],$$

and $Q^{\pi}(s,a)$ with:

$$L(\theta) = \mathbb{E}_{\langle s_t, a_t, r_t, s_{t+1}\rangle \sim U(D)}\left[\left(r_t + \gamma V(s_{t+1};\phi^{-}) - Q(s_t,a_t;\theta)\right)^2\right],$$

A Novel Family of Deep Reinforcement Learning Algorithms

... or by learning $V^{\pi}(s;\phi)$ with

$$L(\phi) = \mathbb{E}_{\langle s_t, a_t, r_t, s_{t+1}\rangle \sim U(D)}\left[\left(r_t + \gamma \max_{a \in \mathcal{A}} Q(s_{t+1}, a; \theta^{-}) - V(s_t;\phi)\right)^2\right],$$

and $Q^{\pi}(s,a)$ with:

$$L(\theta) = \mathbb{E}_{\langle s_t, a_t, r_t, s_{t+1}\rangle \sim U(D)}\left[\left(r_t + \gamma V(s_{t+1};\phi) - Q(s_t, a_t;\theta)\right)^2\right].$$

A Novel Family of Deep Reinforcement Learning Algorithms

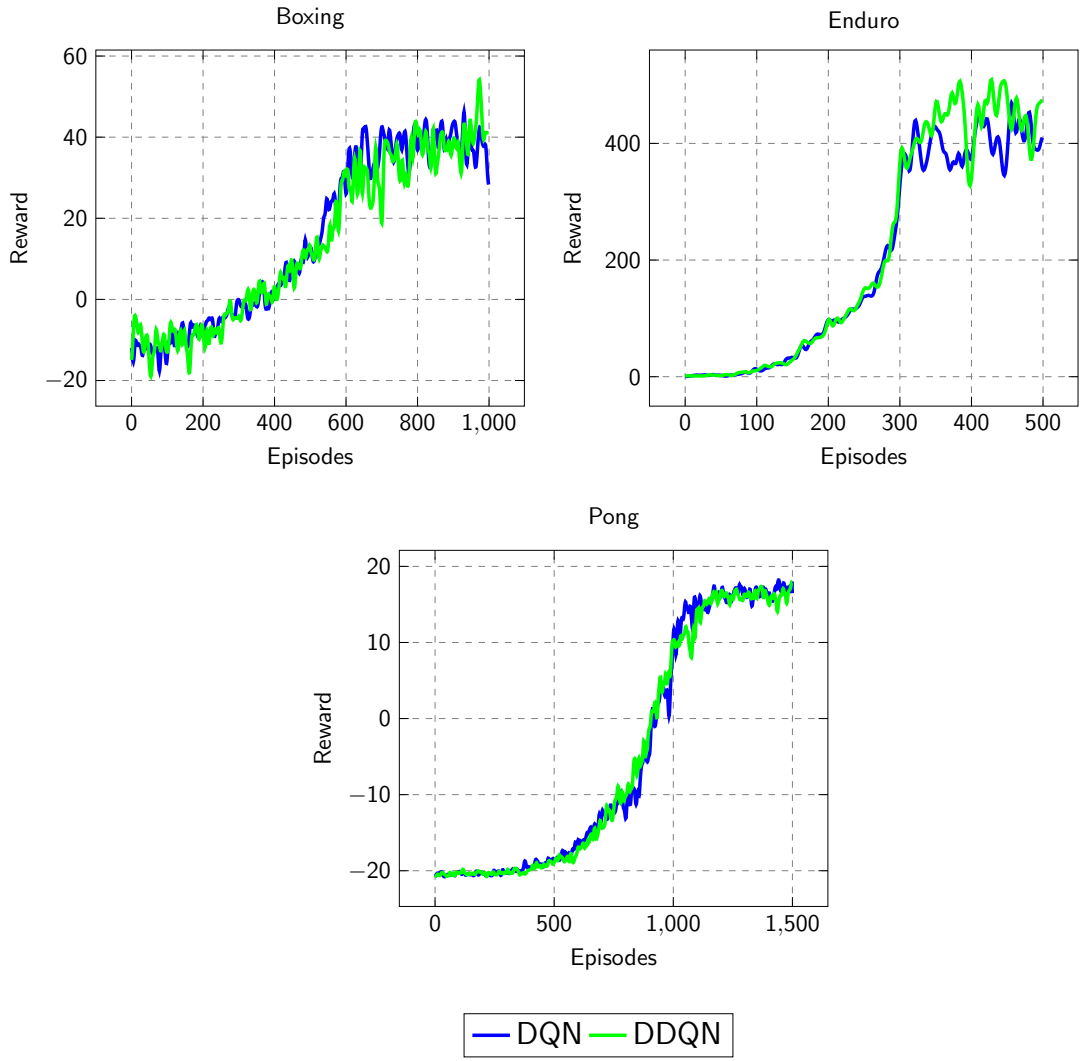
The first two objective functions constitute the **DQV-Learning** algorithm, whereas the second two objective functions define the **DQV-Max Learning** algorithm.

We compare their performance to algorithms which only learn an approximation of the state-action value function $Q^{\pi}(s, a)$ [3] [4]

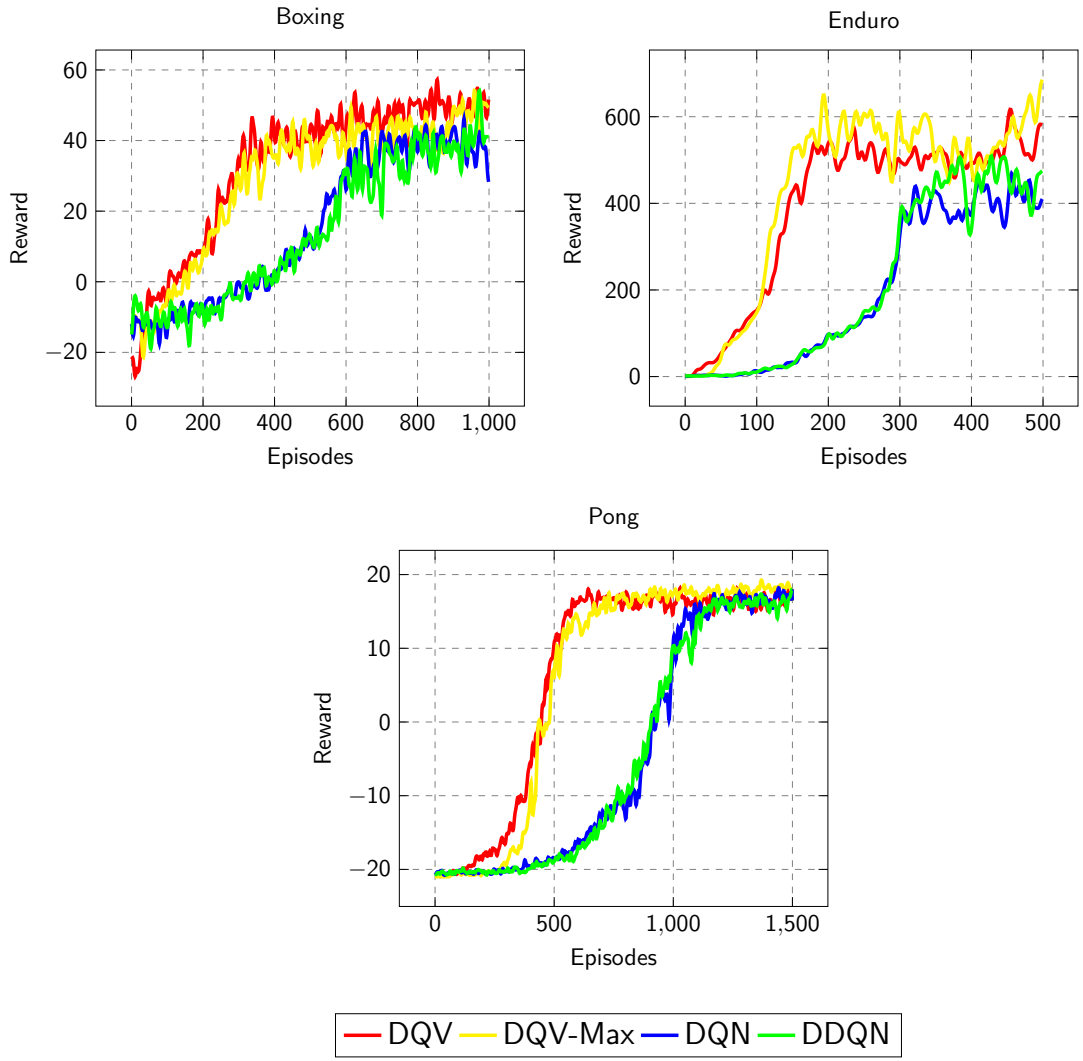
[3] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." nature 518.7540 (2015)

[4] Van Hasselt, Hado, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning." Proceedings of the AAAI conference on artificial intelligence (2016).

A Novel Family of Deep Reinforcement Learning Algorithms

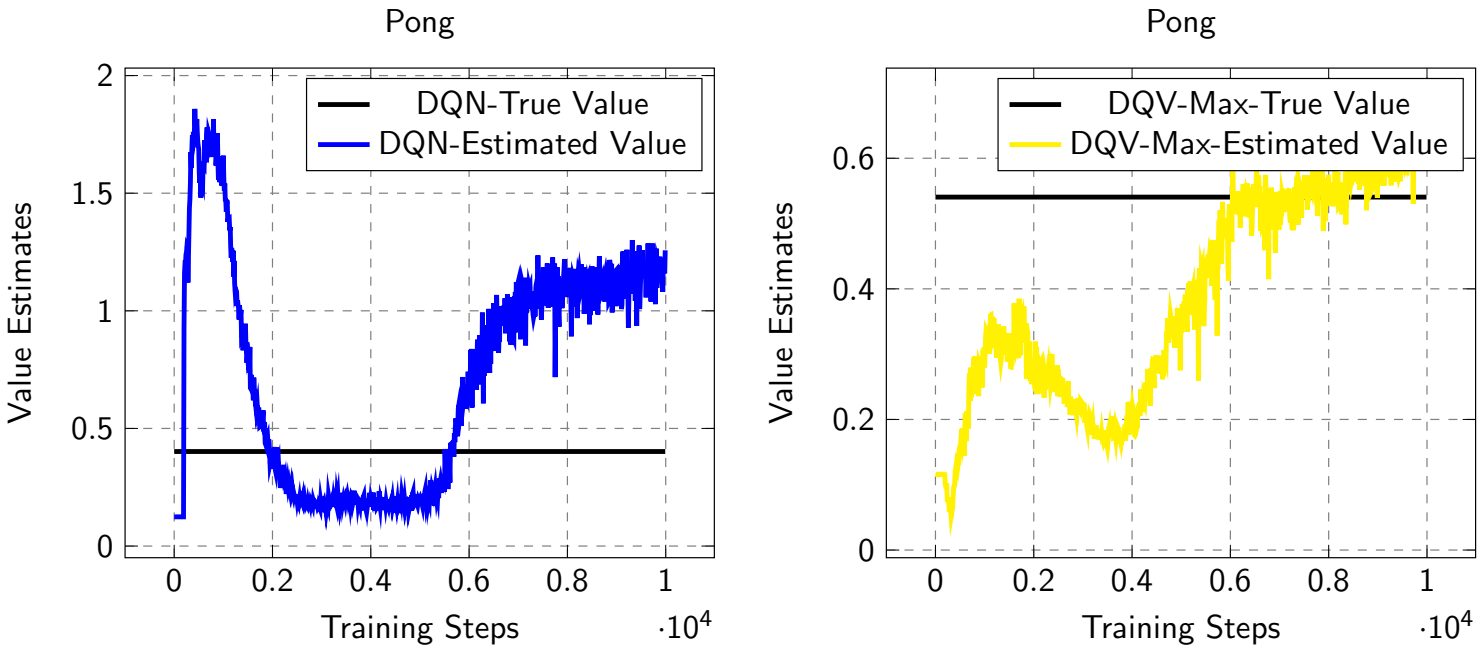


A Novel Family of Deep Reinforcement Learning Algorithms



A Novel Family of Deep Reinforcement Learning Algorithms

$\Rightarrow$ Jointly approximating two value functions over one results in faster convergence, but also in accurate value estimates:



A Novel Family of Deep Reinforcement Learning Algorithms

To conclude DQV-Learning and DQV-Max Learning

- Result in faster learning
- Are robust to the "Deadly Triad of DRL"
- Suffer less from the overestimation bias of the Q function
- Scale well to the multi-agent setting [5]

However ...

- Memory wise are twice as expensive as DQN and DDQN
- Do not always result in better policies

[5] Leroy, Pascal, et al. "QVMix and QVMix-Max: Extending the Deep Quality-Value Family of Algorithms to Cooperative Multi-Agent Reinforcement Learning." (2021).

On the Transferability of Deep-Q Networks

On the Transferability of Deep-Q Networks

Let's take a look at the performance of the DQV algorithms more closely ...

Environment	Random	Human	DQN	DDQN	DQV	DQV-Max
Asteroids	719.10	13156.70	1629.33	930.60	1445.40	1846.08
Bank Heist	14.20	734.40	429.67	728.30	1236.50	1118.28
Boxing	0.10	4.30	71.83	81.70	78.66	80.15
Crazy Climber	10780.50	35410.50	114103.33	101874.00	108600.00	1000131.00
Enduro	0.00	309.60	301.77	319.50	829.33	875.64
Fishing Derby	-91.70	5.50	-0.80	20.30	1.12	20.42
Frostbite	65.20	4334.70	328.33	241.50	271.86	281.36
Gopher	257.60	2321.00	8520.00	8215.40	8230.30	7940.00
Ice Hockey	-11.20	0.90	-1.60	-2.40	-1.88	-1.12
James Bond	29.00	406.70	576.67	438.00	372.41	440.80
Montezuma's Revenge	0.00	4366.70	0.00	0.00	0.00	0.00
Ms.Pacman	307.30	15693.40	2311.00	3210.00	3590.00	3390.00
Pong	-20.70	9.30	18.90	21.00	21.00	21.00
Road Runner	11.50	7845.00	18256.67	48377.00	39290.00	20700.00
Zaxxon	32.50	9173.30	4976.67	10182.00	10950.00	8487.00

On the Transferability of Deep-Q Networks

Based on these results we have formulated the following research questions:

- Can we improve the performance of DQV and DQV-Max Learning through Transfer Learning?
- Is parametric transfer as effective in the model-free Reinforcement Learning setting as it is in the Supervised Learning setting?

On the Transferability of Deep-Q Networks

$\Rightarrow$ We study the Transfer Learning potential of several types of Deep-Q Networks

- We start by considering two variants of Deep-Q Networks: the DDQN and DQV-Learning algorithms
- The popular `Atari` games are used as benchmark
- We explore what happens when models that are pre-trained on $\mathcal{M}_S$ get fine-tuned on $\mathcal{M}_T$

On the Transferability of Deep-Q Networks

Some examples of source and target tasks ...

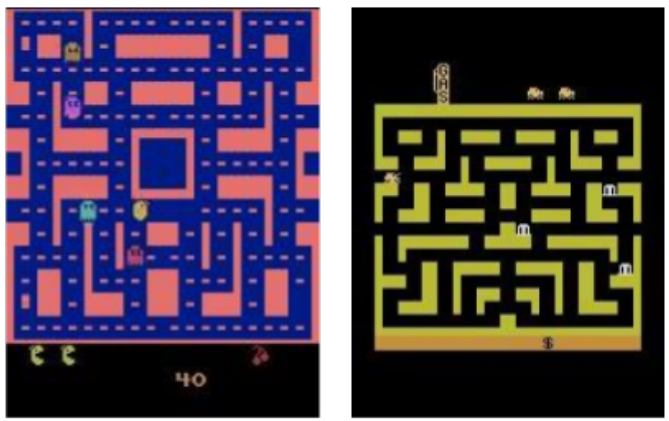


Figure: The visually similar `Ms Pacman` and `Bank Heist` games.

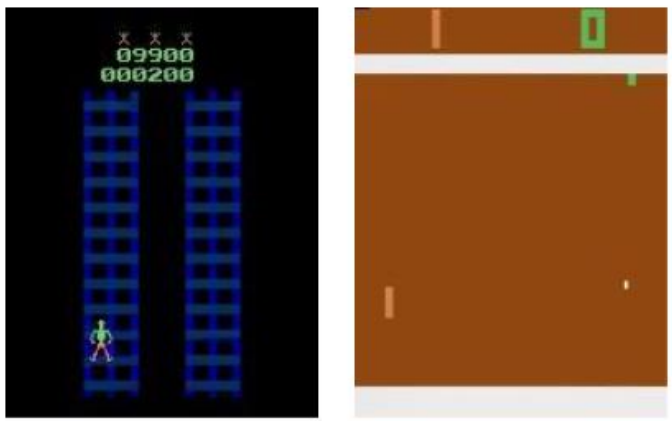


Figure: The highly different `Crazy Climber` and `Pong` games.

On the Transferability of Deep-Q Networks

We quantitatively measure the benefits of transferring and fine-tuning a pre-trained model by computing the area ratio metric [6] $\mathcal{R}$

$$\mathcal{R} = \frac{\text{area of } \mathcal{M}_S - \text{area of } \mathcal{M}_T}{\text{area of } \mathcal{M}_T}$$

expecting mostly positive values for $\mathcal{R}$

[6] Lazaric, Alessandro. "Transfer in reinforcement learning: a framework and a survey." Reinforcement Learning. Springer (2012).

On the Transferability of Deep-Q Networks

However, we noticed that adopting Transfer Learning strategies performed surprisingly bad

DQV	BankHeist	Boxing	CrazyClimber	Enduro	FishingDerby	Frostbite	JamesBond	MsPacman	Pong	Zaxxon
BankHeist	-	-0.019	-1	-0.317	0.5	0.729	0.973	-0.089	-1.238	-0.998
Boxing	-0.494	-	-0.278	-0.852	0.552	-0.01	0.247	-0.184	-0.841	-0.999
CrazyClimber	-0.569	-0.261	-	-0.593	0.19	0.277	0.621	-0.111	-1.206	-0.178
Enduro	-0.571	-0.018	-0.25	-	0.726	-0.017	-0.41	-0.08	-0.466	-0.164
FishingDerby	-1	-0.893	-0.093	-0.45	-	0.068	0.197	-0.136	-3.083	-0.999
Frostbite	-0.933	0.024	-1	-0.348	0.222	-	0.569	0.009	-0.663	-0.076
JamesBond	-0.123	-0.106	-0.131	-0.033	0.519	0.262	-	0.218	-1.329	-1
MsPacman	-0.985	-0.219	-0.012	-0.494	0.6	0.346	0.398	-	-1.646	-0.997
Pong	-1	-0.083	-0.428	-0.476	0.725	-0.024	0.896	0.123	-	-0.729
Zaxxon	-0.76	-0.028	0.037	-0.116	0.385	0.16	-0.253	0.06	-1.602	-

DDQN	BankHeist	Boxing	CrazyClimber	Enduro	FishingDerby	Gopher	IceHockey	Jamesbond	MsPacman	Pong
BankHeist	-	0.121	-0.378	-0.006	-0.107	0.042	-0.006	-0.058	0.001	-3.013
Boxing	-0.316	-	-0.104	-0	0.038	0.06	0.015	-0.225	-0.027	0.936
CrazyClimber	-0.192	-0.487	-	-0.012	-0.084	0.016	0.015	0.016	-0.015	-2.64
Enduro	-0.296	0.193	-0.167	-	0.039	0.03	0.019	-0.235	-0.039	0.248
FishingDerby	-0.212	-0.545	-1	-0.085	-	0.016	0.001	-0.055	-0.026	-0.935
Gopher	-0.466	0.044	-0.108	-0.005	0.007	-	-0.005	-0.094	-0.02	-1.816
IceHockey	-0.046	0.245	-0.067	0.014	-0.178	0.072	-	0.037	-0.015	0.112
Jamesbond	-0.145	0.232	-0.064	0.005	-0.267	0.031	-0.092	-	-0.006	-1.578
MsPacman	-0.173	-1.179	-0.129	-0.06	0.003	-0.019	0.007	0.071	-	-2.774
Pong	-0.127	0.028	-0.12	0.01	0.037	0.042	0.002	-0.174	-0.006	-

On the Transferability of Deep-Q Networks

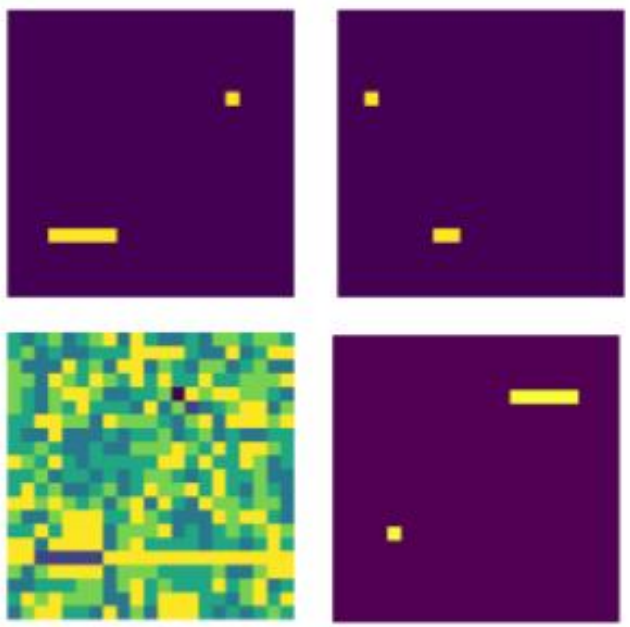
Specifically, on the `Atari` benchmark we have noticed that:

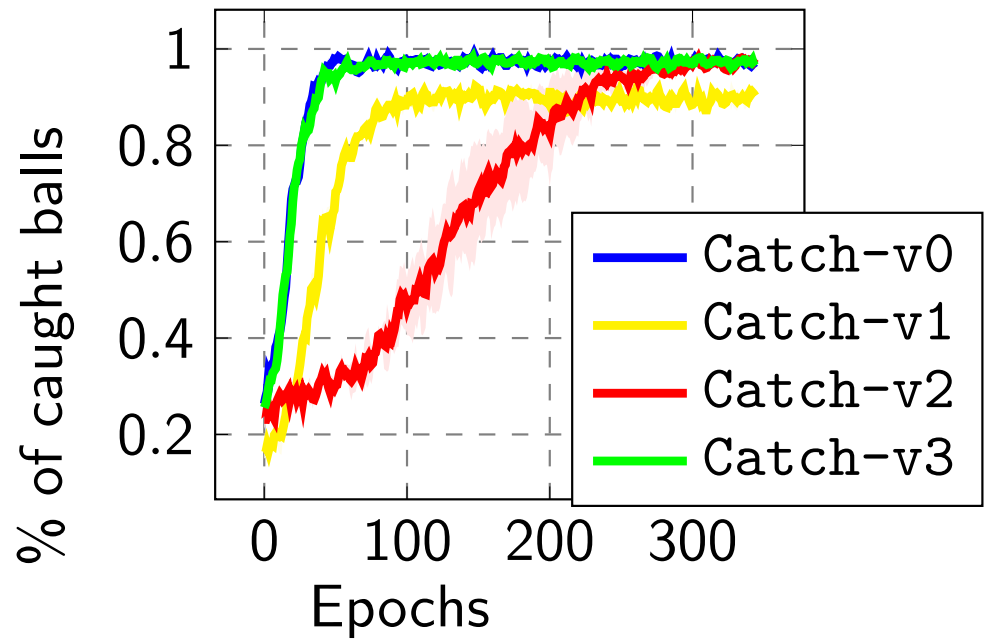
- Transferring pre-trained Deep-Q Networks mostly results in negative transfer, even across similar games `Bank Heist` $\rightarrow$ `Ms. Pacman`
- Transfer across environments is not symmetric: `Bank Heist` $\leftrightarrow$ `Fishing Derby`
- Different algorithms result in different transfer learning performance

$\Rightarrow$ Can we explain these results any further?

On the Transferability of Deep-Q Networks

We have designed a set of novel control tasks that could help us better understanding the transfer learning potential of Deep-Q Networks





On the Transferability of Deep-Q Networks

$\Rightarrow$ Even on our newly designed problems, we keep observing very poor transfer learning performance

Table: The area ratio obtained after fine-tuning a pre-trained DQN agent on the different `Catch` environments. We can see that no matter which source game is used for pre-training, transfer learning surprisingly never results in positive transfer.

	Catch-v0	Catch-v2	Catch-v3	Catch-v4
Catch-v0	-	-0.026	-0.486	-0.479
Catch-v2	-0.16	-	-0.121	-0.248
Catch-v3	-0.406	-0.313	-	-0.465
Catch-v4	-0.016	-0.24	-0.179	-

On the Transferability of Deep-Q Networks

We therefore asked ourselves whether Deep-Q Networks are at least able to self-transfer ...

- What happens when the source task $\mathcal{M}_S$ and the target task $\mathcal{M}_T$ are identical?
- Are there any differences between using an off-the-shelf approach compared to a fine-tuning strategy?
- Let's recall that when transferring a pre-trained model only the last layer responsible for estimating $Q(s, a)$ is randomly initialized

On the Transferability of Deep-Q Networks

We notice that positive self-transfer can get obtained as long as the networks do not get fine-tuned but are used as feature extractors instead

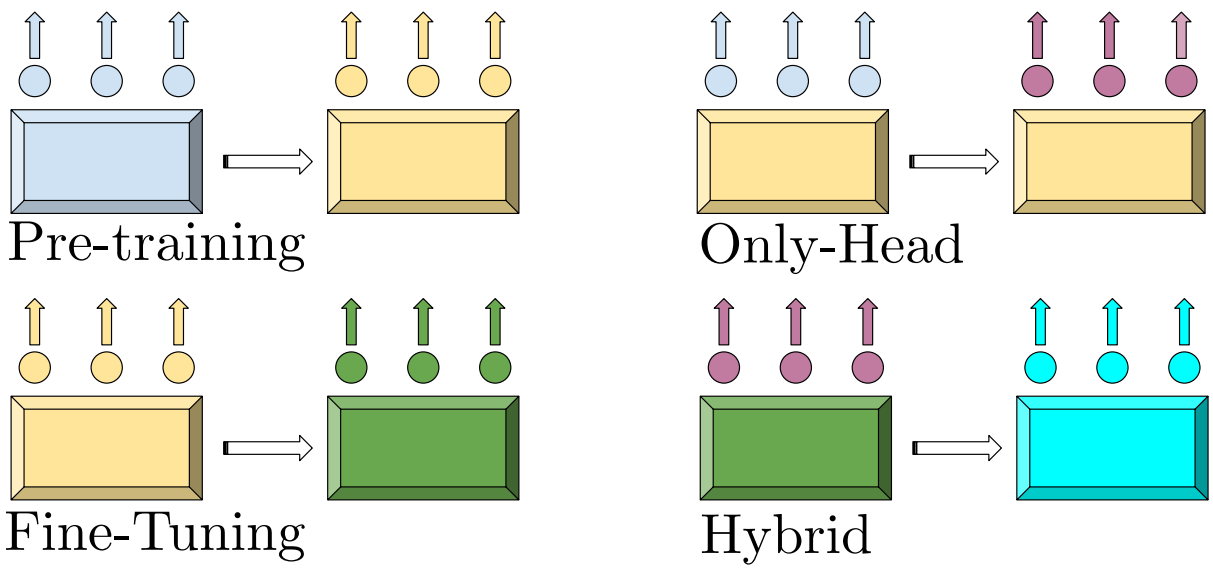
Table: The area ratio scores obtained after performing self-transfer. We can see that if only the last linear layer is trained, then positive transfer is obtained on all `Catch` environments, whereas if the network is fine-tuned, positive transfer is (in part) only obtained on `Catch-v2`.

	Catch-v0	Catch-v1	Catch-v2	Catch-v3
Only-Head	0.05	0.141	0.674	0.059
Fine-Tuning	0.017	-0.218	0.393	-0.236

On the Transferability of Deep-Q Networks

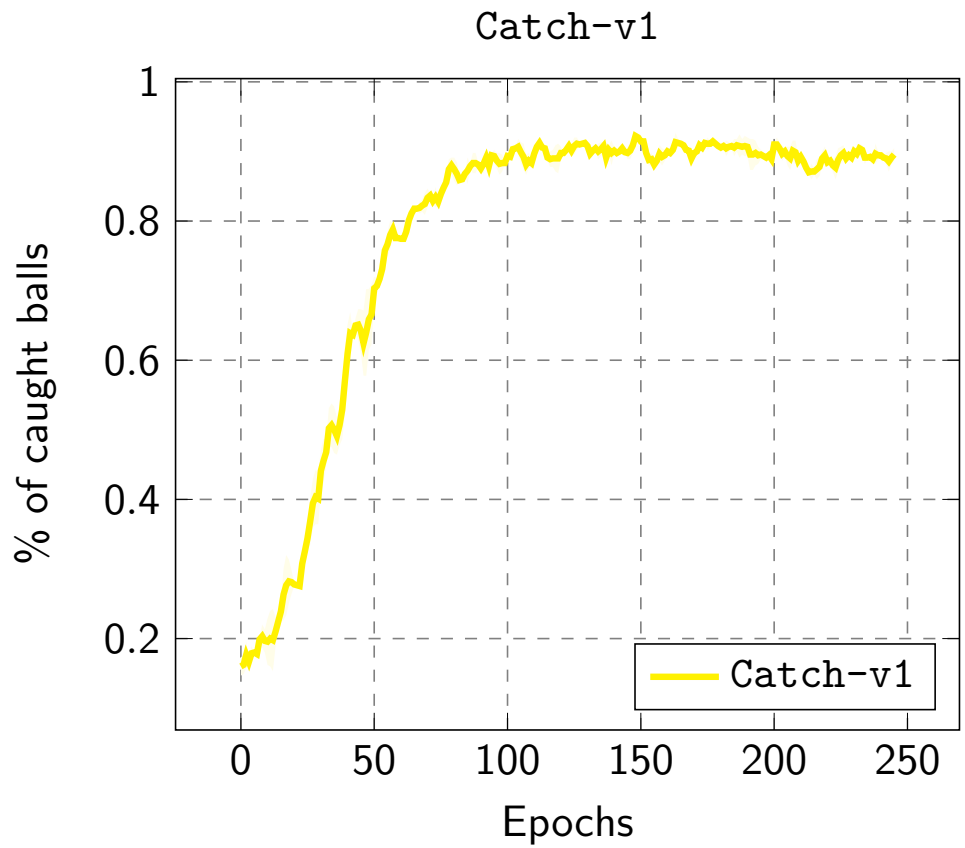
Our conjecture is that a Deep-Q Network can only solve a problem if it has carefully found a balance between its feature extractor components and the linear layer that estimates $Q(s, a)$

Let us consider the following Deep-Q Networks



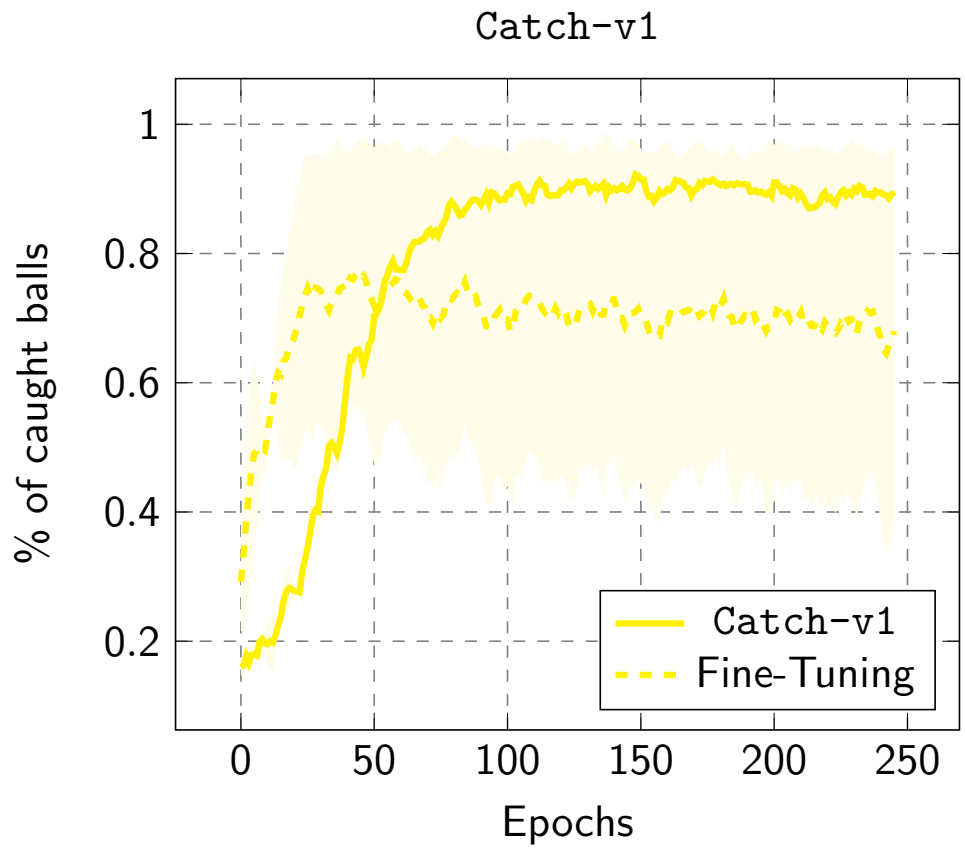
On the Transferability of Deep-Q Networks

$\Rightarrow$ How does this look like in practice?



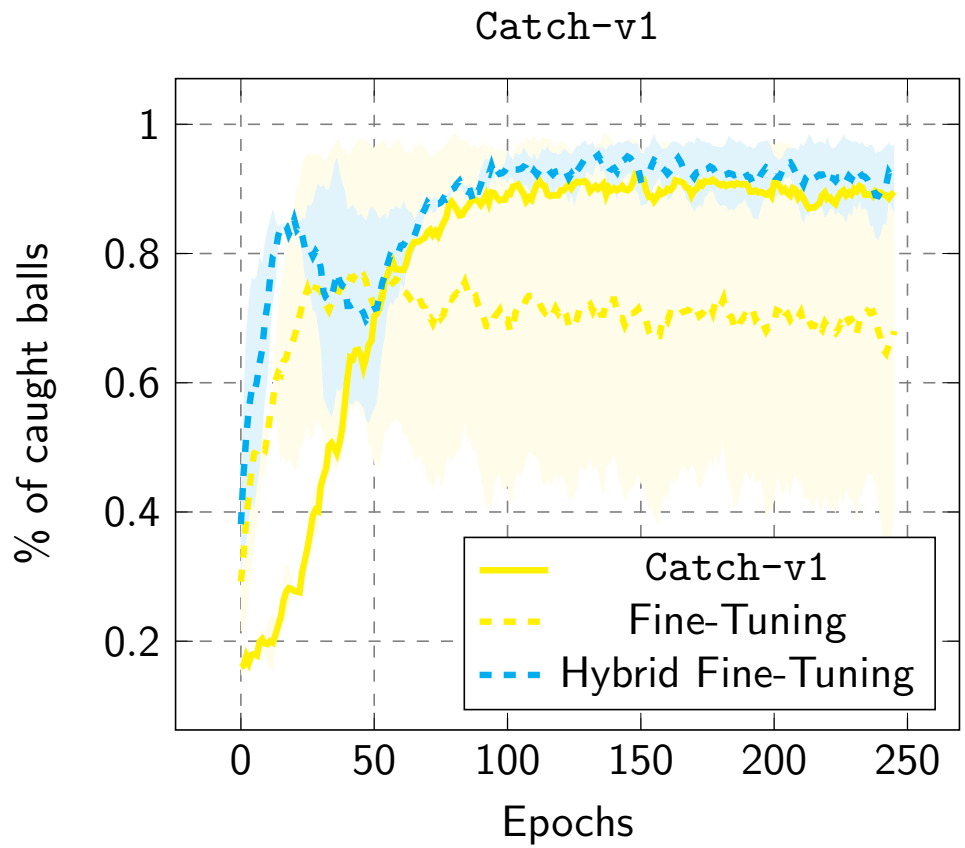
On the Transferability of Deep-Q Networks

$\Rightarrow$ How does this look like in practice?



On the Transferability of Deep-Q Networks

$\Rightarrow$ How does this look like in practice?



On the Transferability of Deep-Q Networks

To conclude PART III ...

- We show that fine-tuning pre-trained models in a model-free Deep Reinforcement Learning context is particularly challenging
- There is a clear difference in terms of performance with the results presented in PART II
- We provide some initial intuition about why negative transfer seems to be happening
- This remains an open problem that we believe deserves attention

The End.

Final References

- Sabatelli, Matthia, et al. "Deep transfer learning for art classification problems." Proceedings of the European Conference on Computer Vision (ECCV) Workshops. 2018.
- Sabatelli, Matthia, et al. "Advances in Digital Music Iconography: Benchmarking the detection of musical instruments in unrestricted, non-photorealistic images from the artistic domain." Digital Humanities Quarterly 15.1 (2021).
- Sabatelli, Matthia, Mike Kestemont, and Pierre Geurts. "On the Transferability of Winning Tickets in Non-natural Image Datasets." VISIGRAPP (5: VISAPP). 2021.
- Sabatelli, Matthia, et al. "Deep Quality Value (DQV) Learning." Deep Reinforcement Learning Workshop of the 32nd Conference on Neural Information Processing Systems (2018).
- Sabatelli, Matthia, et al. "The deep quality-value family of deep reinforcement learning algorithms." 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020.
- Sabatelli, Matthia, and Pierre Geurts. "On The Transferability of Deep-Q Networks." Deep Reinforcement Learning Workshop of the 35th Conference on Neural Information Processing Systems. 2021.