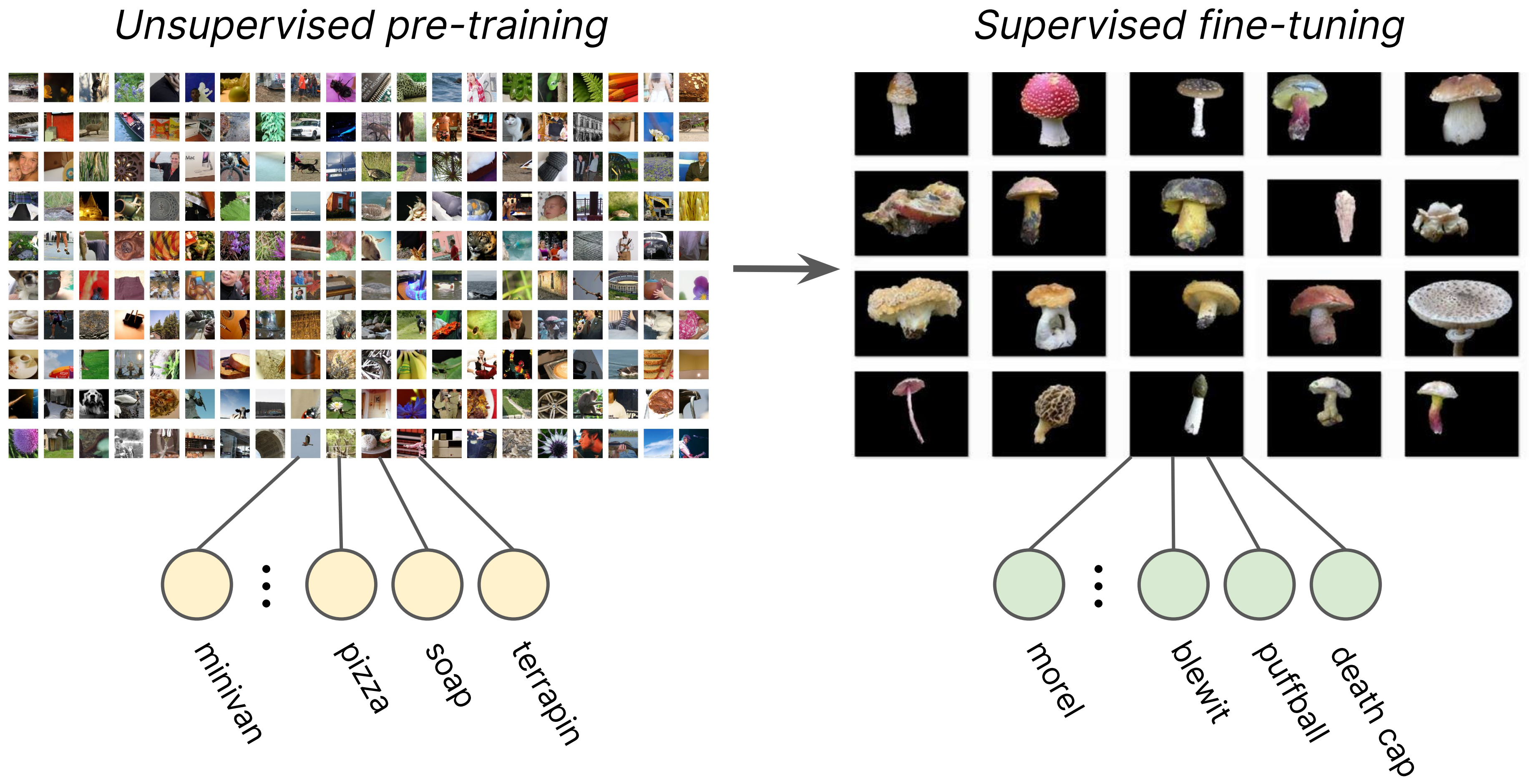
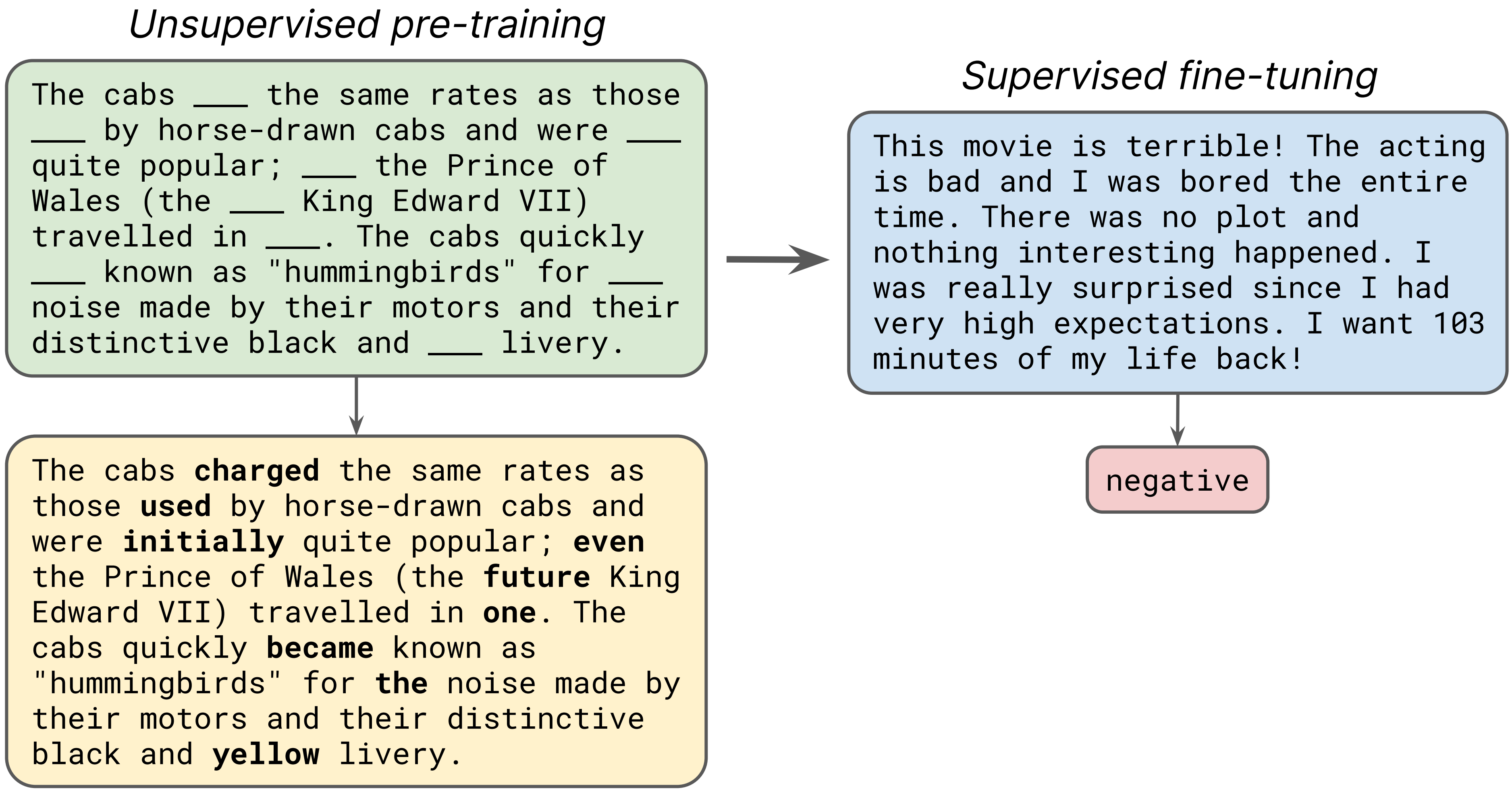
A call to build models like we build open-source software

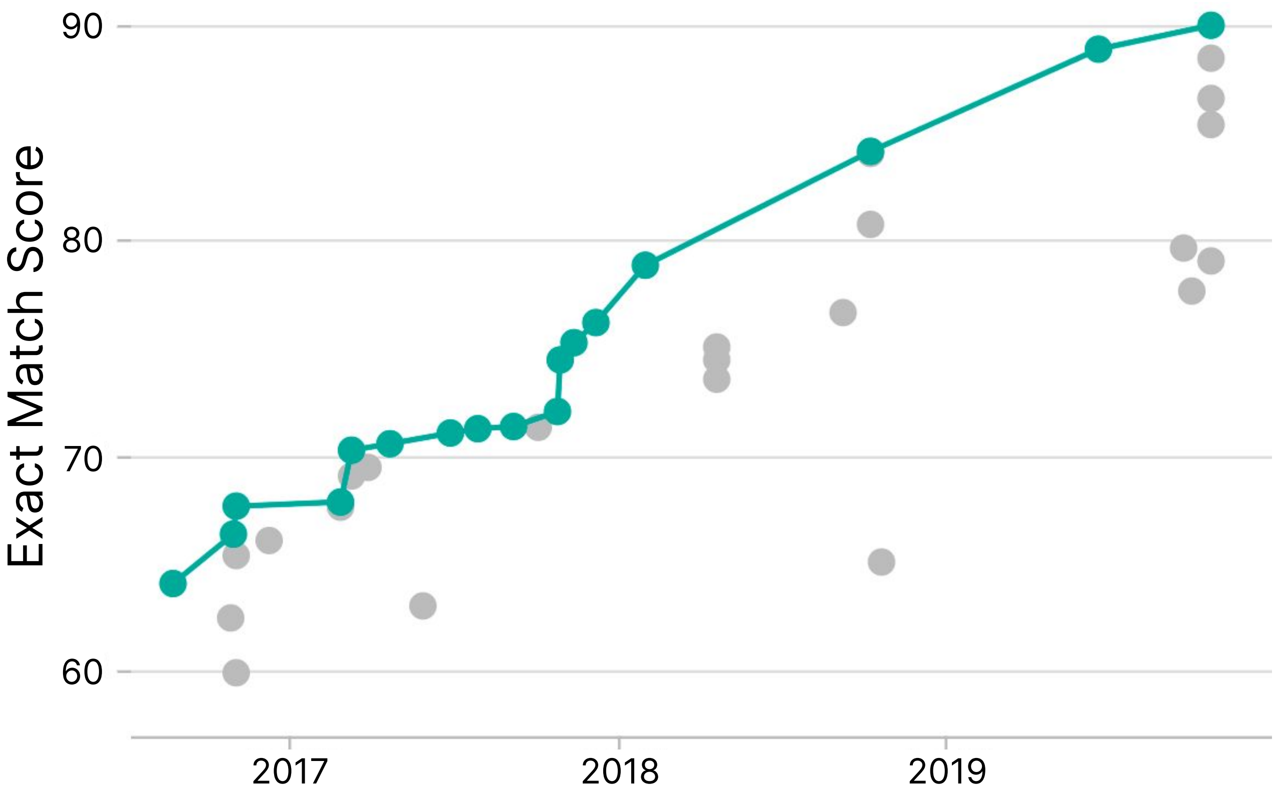
Colin Raffel



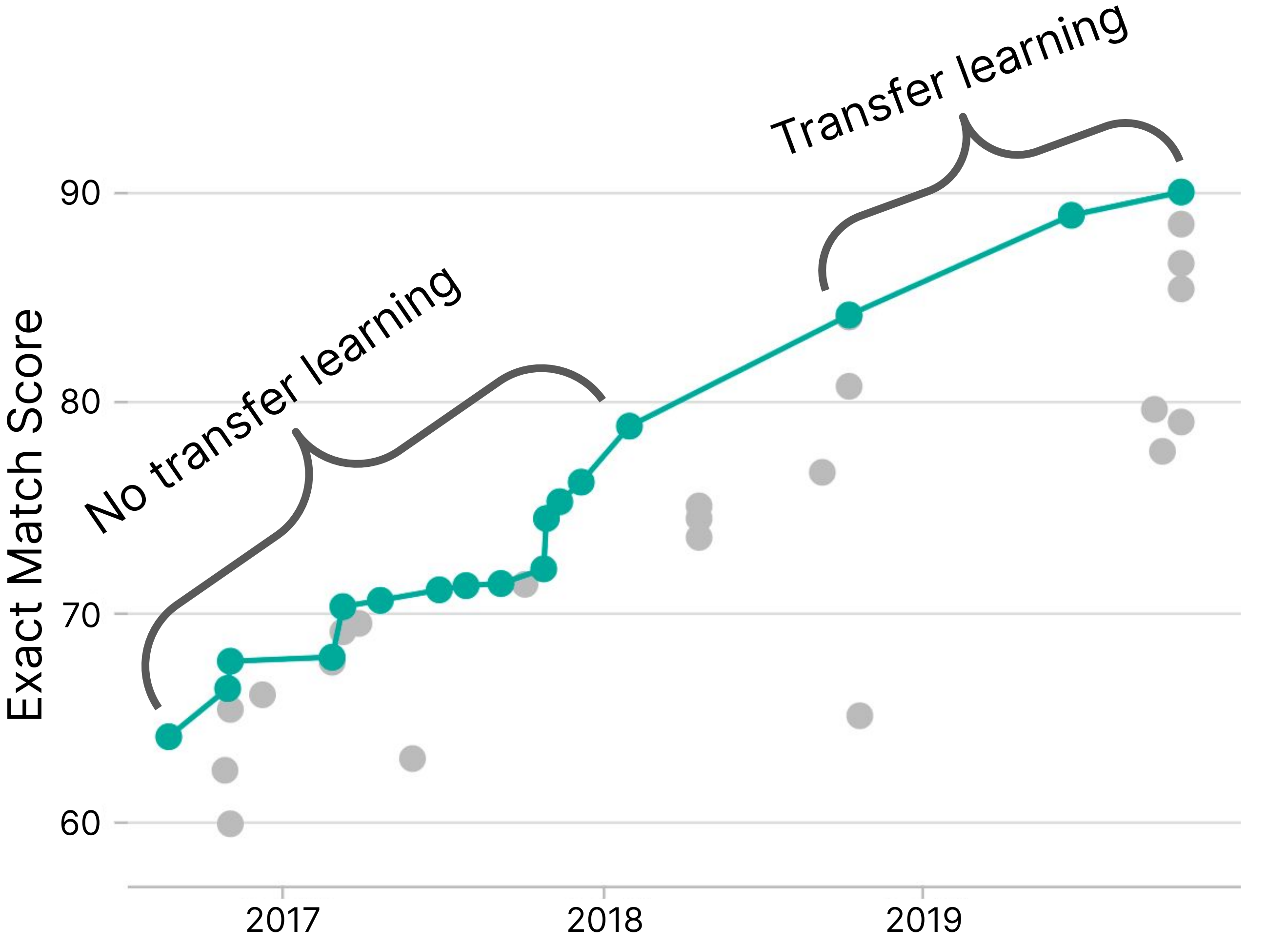
Unsupervised pre-training
Supervised fine-tuning
minivan
pizza
soap
terrapin
morel
blewit
puffball
death cap


Unsupervised pre-training
The cabs ___ the same rates as those ___ by horse-drawn cabs and were ___ quite popular; ___ the Prince of Wales (the ___ King Edward VII) travelled in ___. The cabs quickly ___ known as "hummingbirds" for ___ noise made by their motors and their distinctive black and ___ livery.
The cabs **charged** the same rates as those **used** by horse-drawn cabs and were **initially** quite popular; **even** the Prince of Wales (the **future** King Edward VII) travelled in **one**. The cabs quickly **became** known as "hummingbirds" for **the** noise made by their motors and their distinctive black and **yellow** livery.
Supervised fine-tuning
This movie is terrible! The acting is bad and I was bored the entire time. There was no plot and nothing interesting happened. I was really surprised since I had very high expectations. I want 103 minutes of my life back!
negative

SQuAD Exact Match score

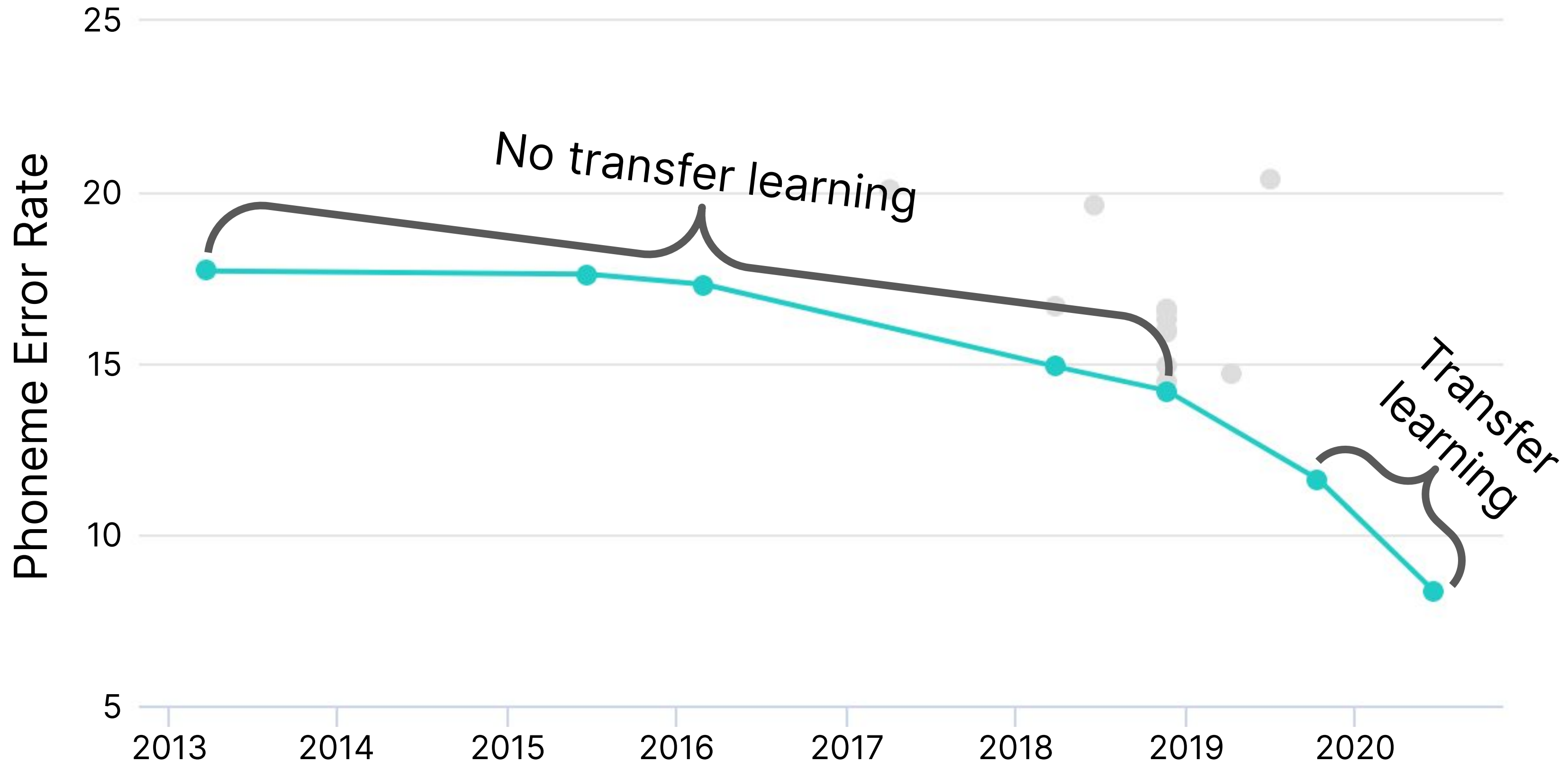


from https://paperswithcode.com/sota/question-answering-on-squad11-dev



from https://paperswithcode.com/sota/question-answering-on-squad11-dev

TIMIT Phoneme Error Rate



from https://paperswithcode.com/sota/speech-recognition-on-timit

TORCHVISION.MODELS

We provide pre-trained models, using the PyTorch `torch.utils.model_zoo`. These can be constructed by passing `pretrained=True`:

```
import torchvision.models as models
resnet18 = models.resnet18(pretrained=True)
alexnet = models.alexnet(pretrained=True)
squeezenet = models.squeezenet1_0(pretrained=True)
vgg16 = models.vgg16(pretrained=True)
densenet = models.densenet161(pretrained=True)
inception = models.inception_v3(pretrained=True)
googlenet = models.googlenet(pretrained=True)
shufflenet = models.shufflenet_v2_x1_0(pretrained=True)
```

*GPT-3 175B model required 3.14E23 FLOPS of computing for training. Even at theoretical 28 TFLOPS for V100 and lowest 3 year reserved cloud pricing we could find, this will take 355 GPU-years and cost **$4.6M** for a single training run.*

from https://lambdalabs.com/blog/demystifying-gpt-3/

🤗 **Hugging Face** Search models, datasets, users...

Models 33,490 | Search Models | Add filters | Sort: Most Downloads

bert-base-uncased
Fill-Mask • Updated May 18 • ↓ 30M • ♡ 54

roberta-large
Fill-Mask • Updated May 21 • ↓ 13.1M • ♡ 20

distilbert-base-uncased
Fill-Mask • Updated Aug 29 • ↓ 4.83M • ♡ 26

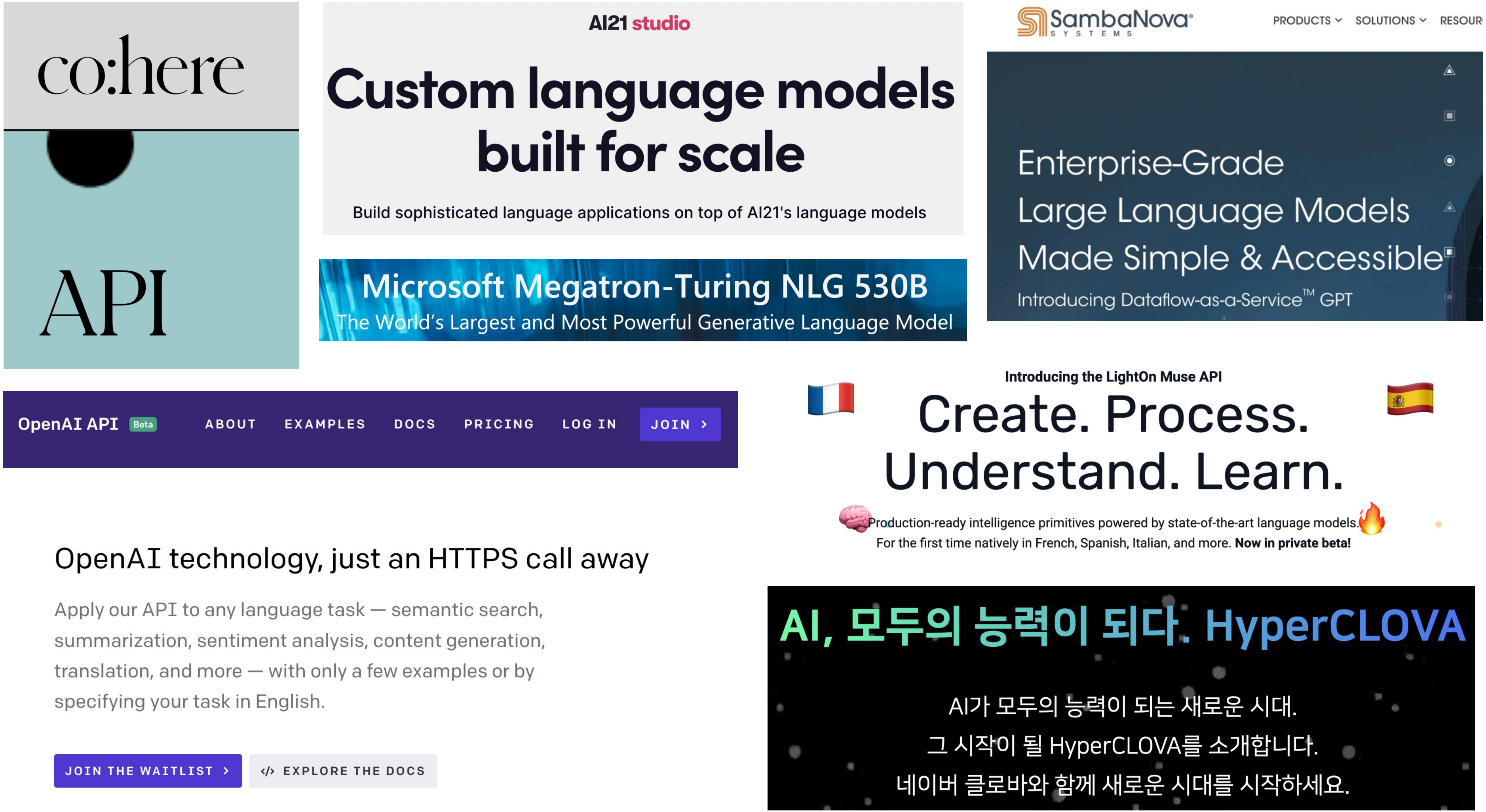
xlm-roberta-base
Fill-Mask • Updated Sep 16 • ↓ 4.78M • ♡ 11

bert-base-cased
Fill-Mask • Updated Sep 6 • ↓ 4.02M • ♡ 6

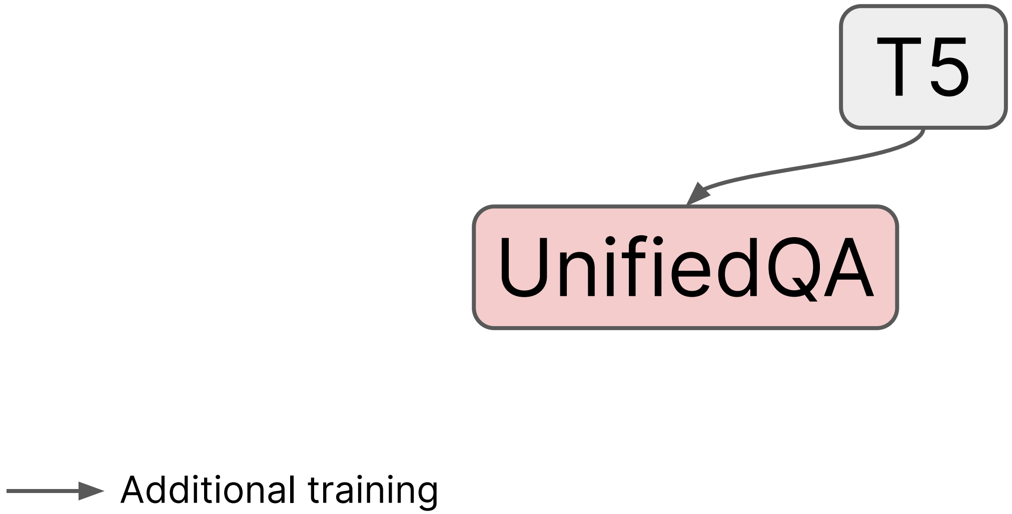
distilbert-base-uncased-finetuned-sst-2-english
Text Classification • Updated Feb 9 • ↓ 3.54M • ♡ 18

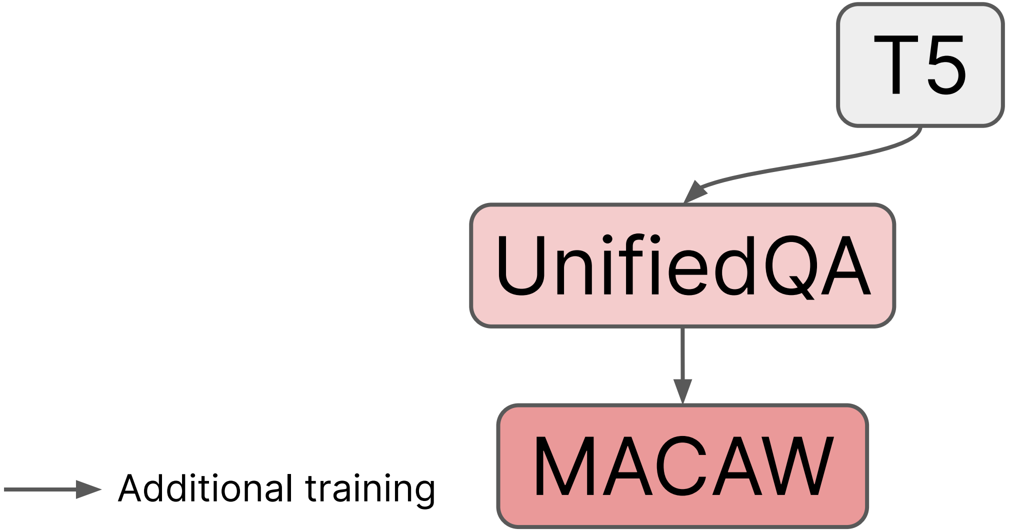
roberta-base
Fill-Mask • Updated Jul 6 • ↓ 3.45M • ♡ 6

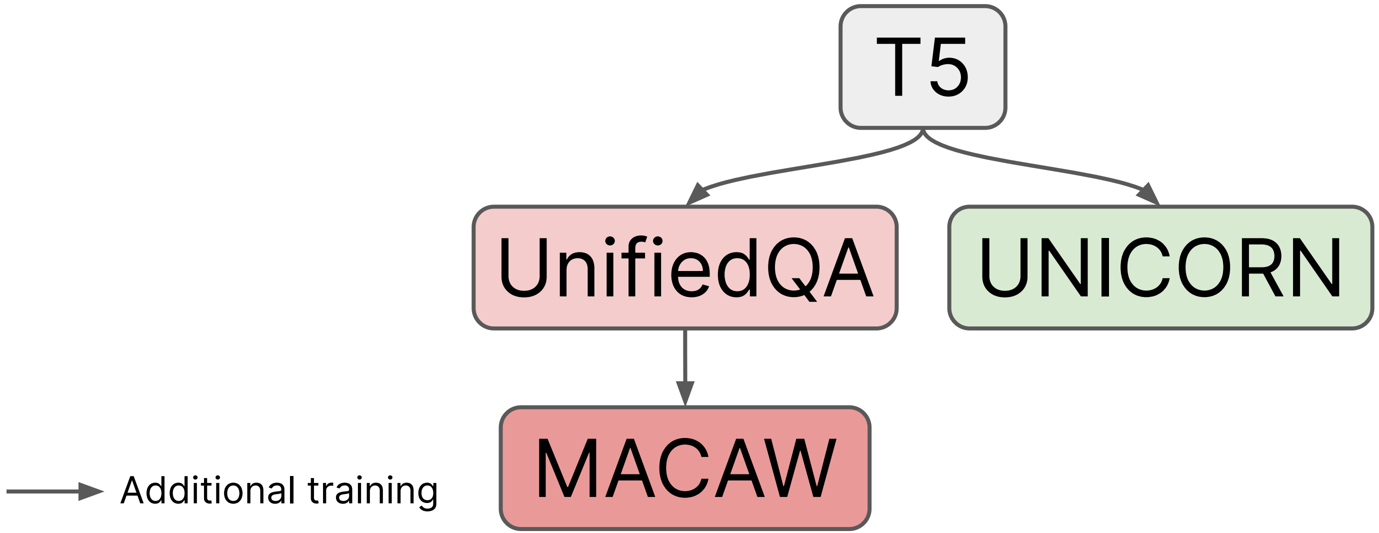
gpt2
Text Generation • Updated May 19 • ↓ 3.34M • ♡ 24

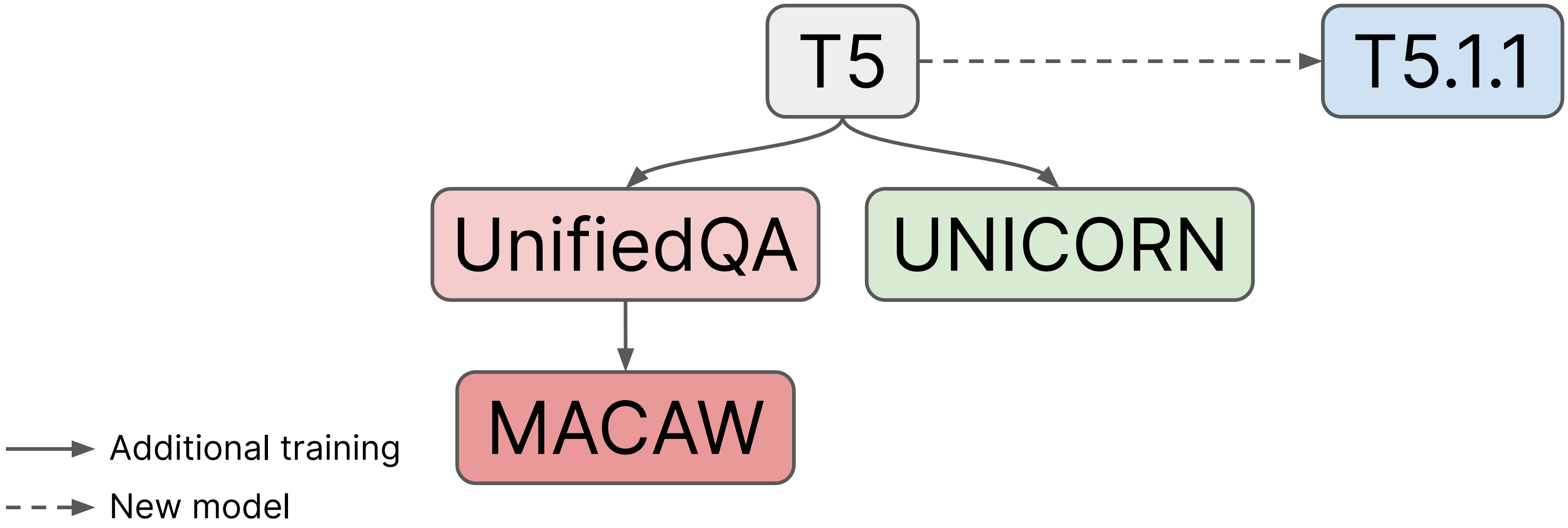

co:here
API
AI21 studio
Custom language models built for scale
Build sophisticated language applications on top of AI21's language models
Microsoft Megatron-Turing NLG 530B
The World's Largest and Most Powerful Generative Language Model
SambaNova SYSTEMS
PRODUCTS
SOLUTIONS
RESOUR
Enterprise-Grade
Large Language Models
Made Simple & Accessible
Introducing Dataflow-as-a-Service™ GPT
OpenAI API Beta
ABOUT
EXAMPLES
DOCS
PRICING
LOG IN
JOIN
OpenAI technology, just an HTTPS call away
Apply our API to any language task — semantic search, summarization, sentiment analysis, content generation, translation, and more — with only a few examples or by specifying your task in English.
JOIN THE WAITLIST
EXPLORE THE DOCS
Introducing the LightOn Muse API
Create. Process. Understand. Learn.
Production-ready intelligence primitives powered by state-of-the-art language models.
For the first time natively in French, Spanish, Italian, and more. Now in private beta!
AI, 모두의 능력이 되다. HyperCLOVA
AI가 모두의 능력이 되는 새로운 시대.
그 시작이 될 HyperCLOVA를 소개합니다.
네이버 클로바와 함께 새로운 시대를 시작하세요.

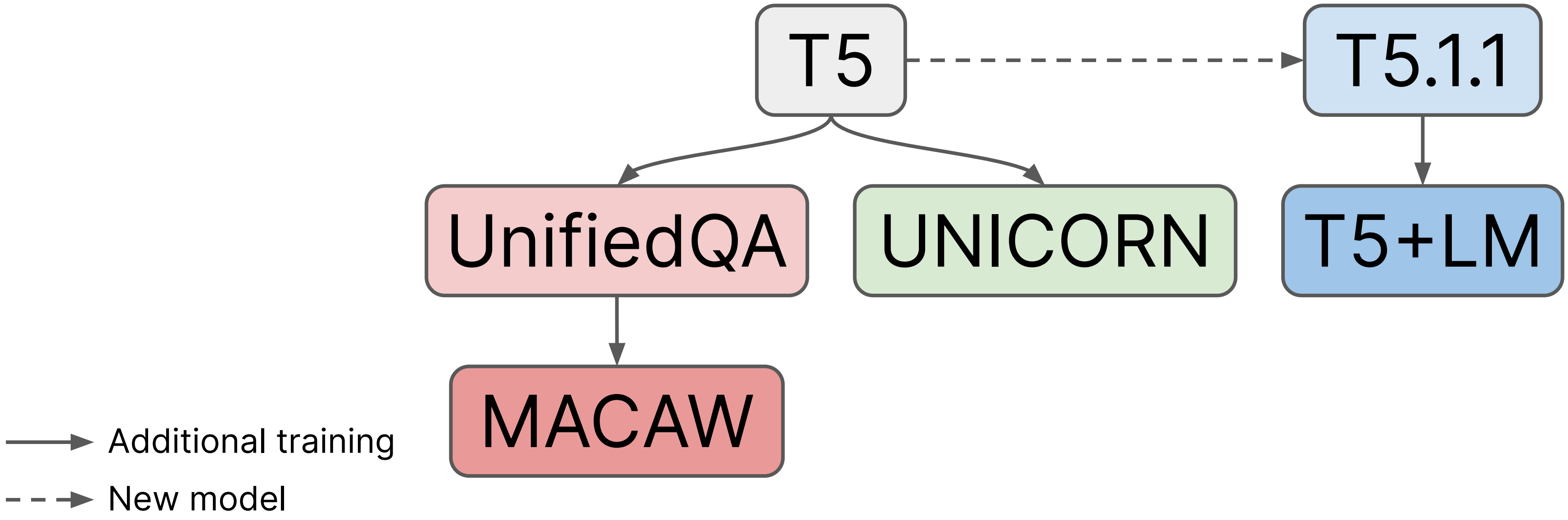
T5

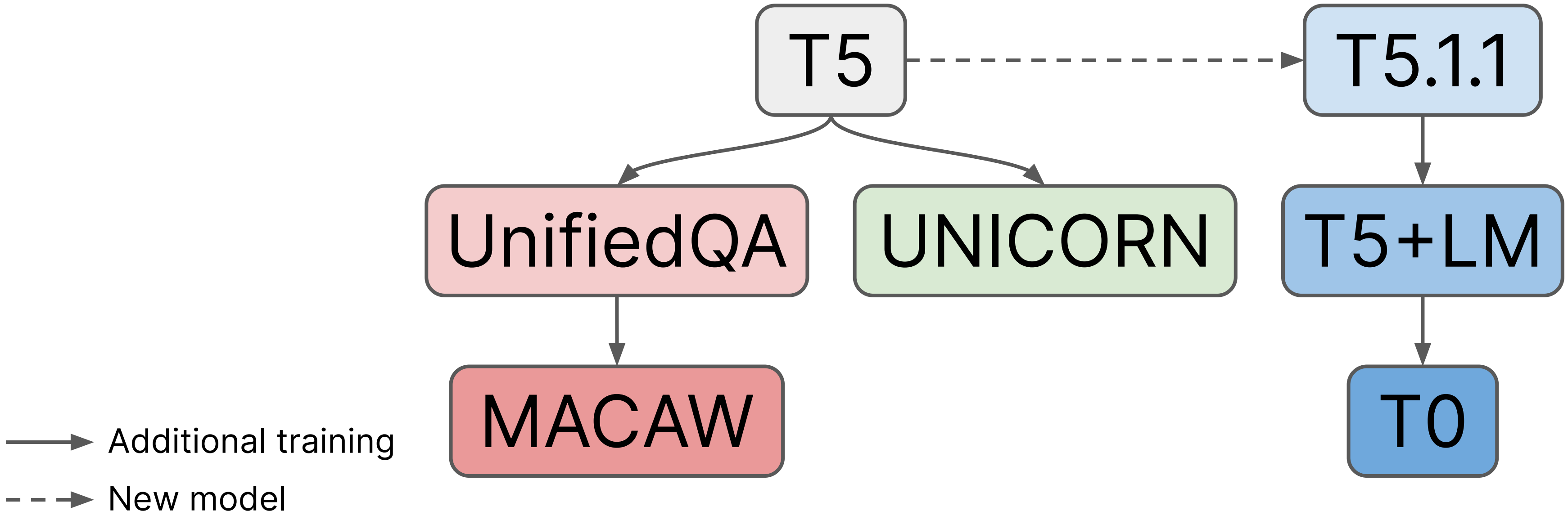

T5
UnifiedQA
Additional training

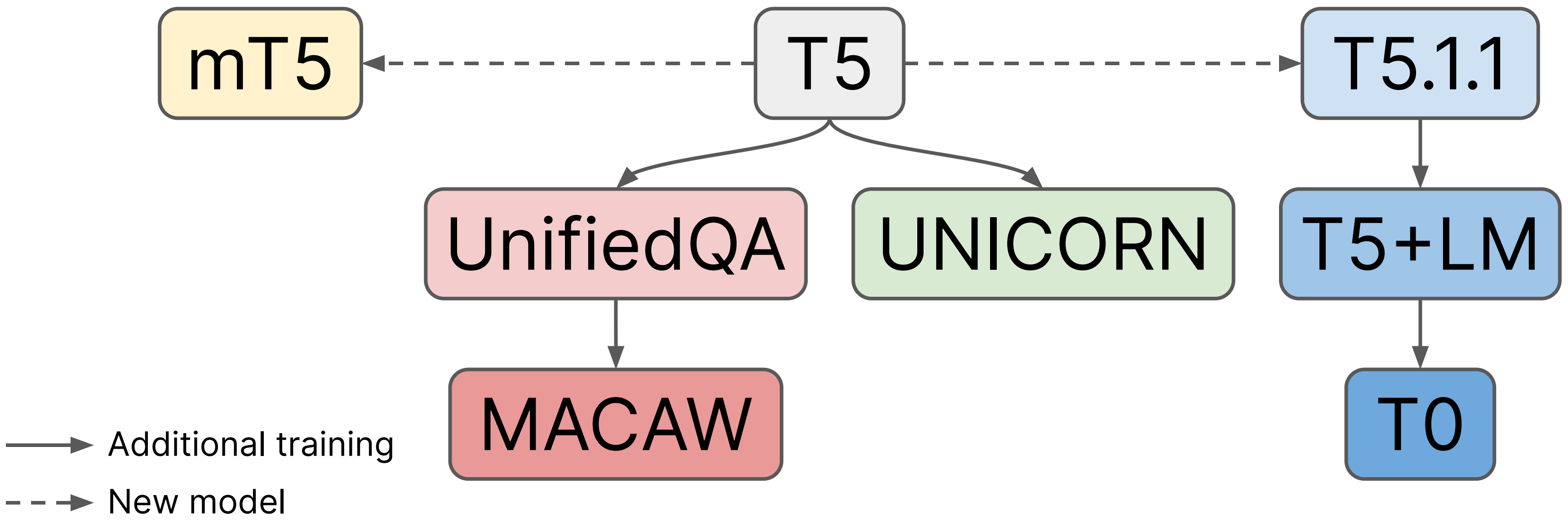

T5
UnifiedQA
MACAW
Additional training

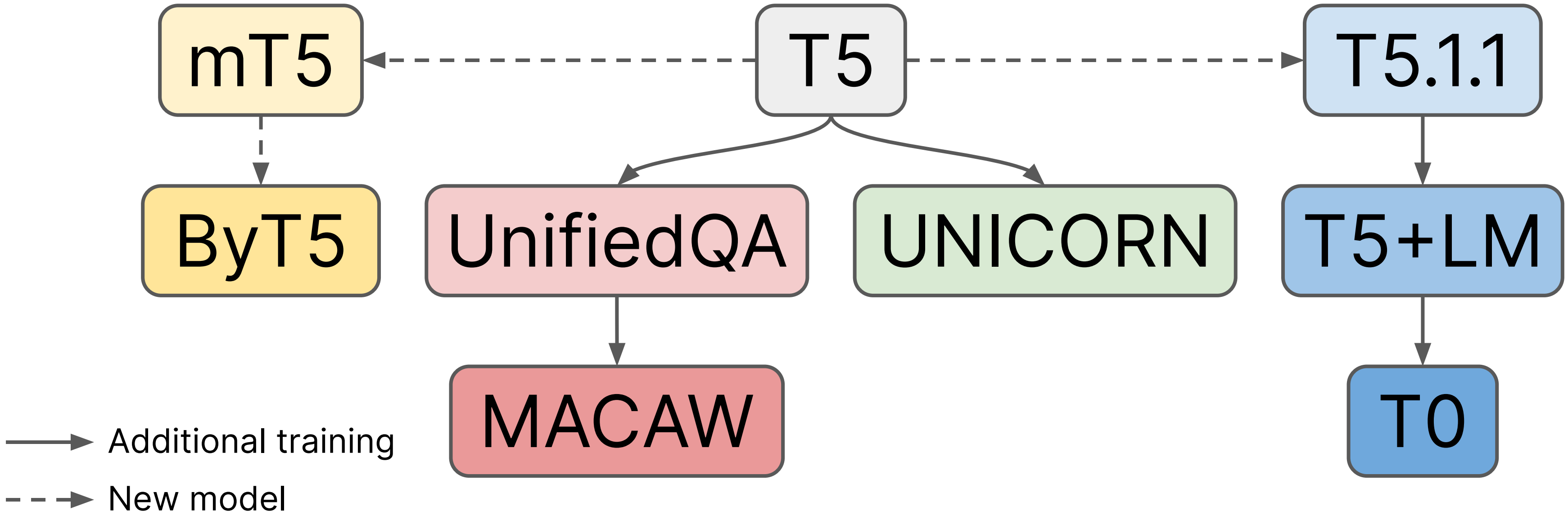

T5
UnifiedQA
UNICORN
MACAW
Additional training

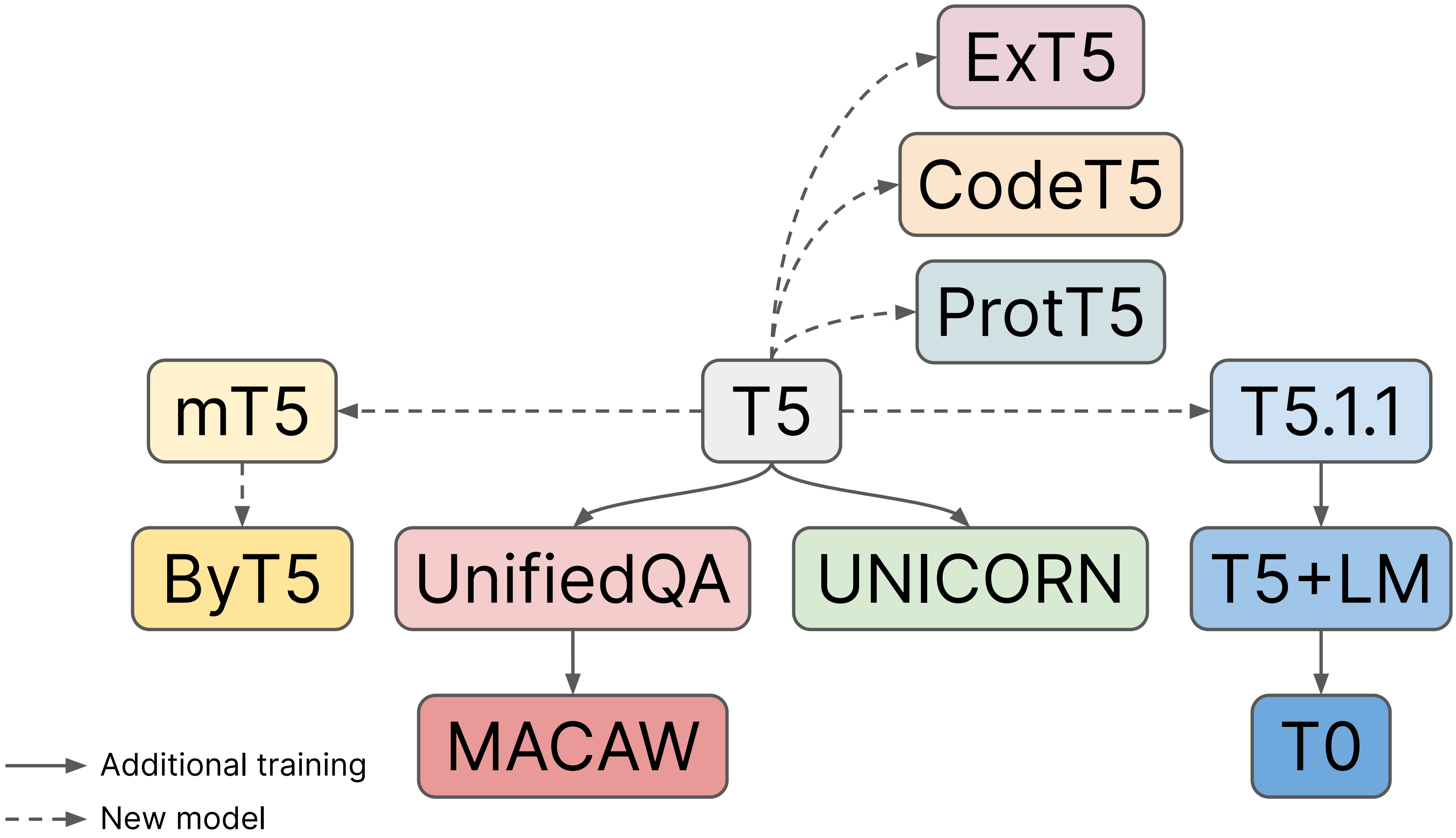

T5
T5.1.1
UnifiedQA
UNICORN
MACAW
Additional training
New model

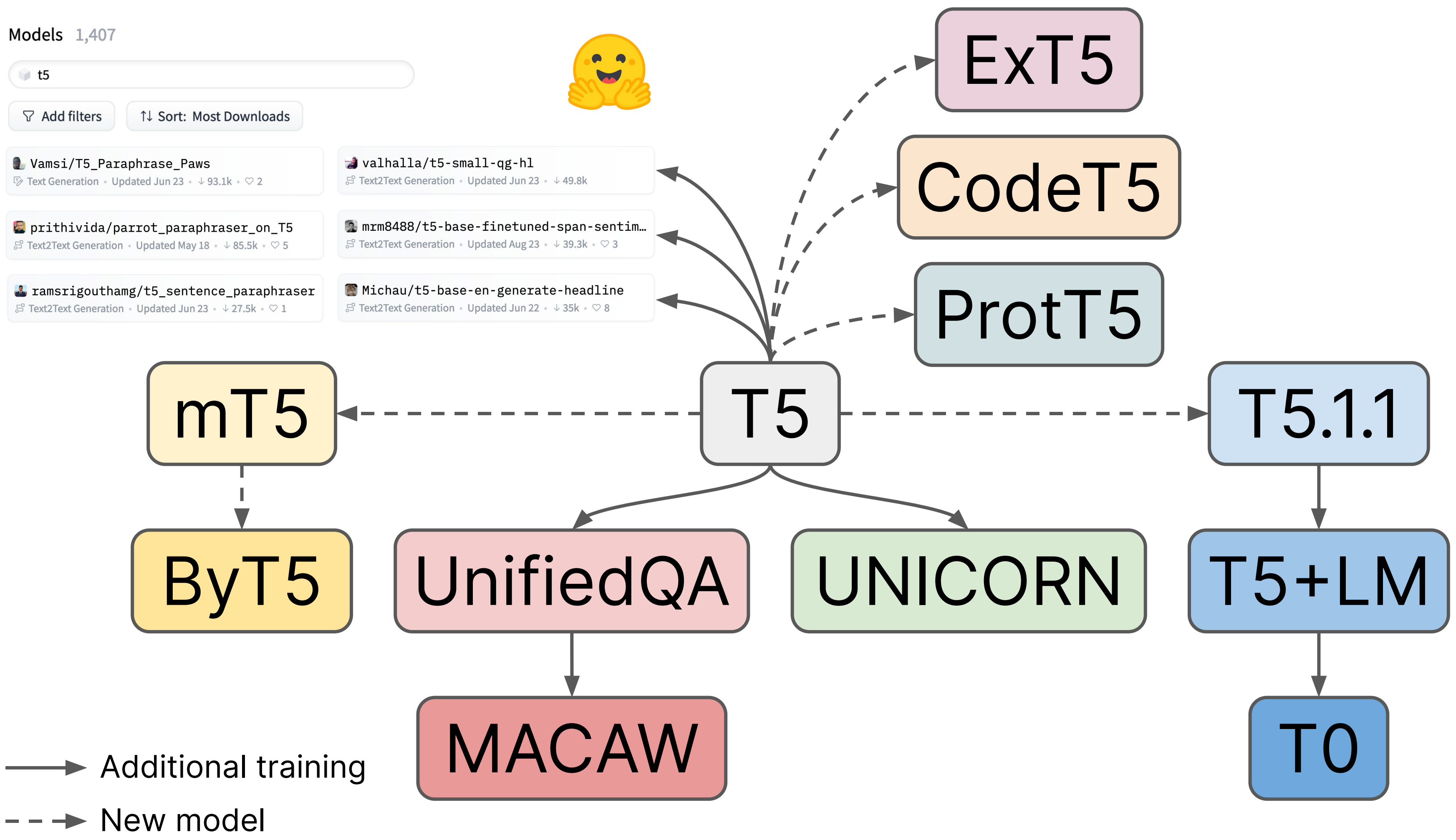

T5
T5.1.1
UnifiedQA
UNICORN
T5+LM
MACAW
Additional training
New model


T5
T5.1.1
UnifiedQA
UNICORN
T5+LM
MACAW
T0
Additional training
New model


mT5
T5
T5.1.1
UnifiedQA
UNICORN
T5+LM
MACAW
T0
Additional training
New model


mT5
T5
T5.1.1
ByT5
UnifiedQA
UNICORN
T5+LM
MACAW
T0
Additional training
New model


ExT5
CodeT5
ProtT5
mT5
T5
T5.1.1
ByT5
UnifiedQA
UNICORN
T5+LM
MACAW
T0
Additional training
New model



Models 1,407
t5
Add filters
Sort: Most Downloads
Vamsi/T5_Paraphrase_Paws
Text Generation • Updated Jun 23 • 93.1k • 2
valhalla/t5-small-qg-hl
Text2Text Generation • Updated Jun 23 • 49.8k
prithivida/parrot_paraphraser_on_T5
Text2Text Generation • Updated May 18 • 85.5k • 5
mrm8488/t5-base-finetuned-span-sentim…
Text2Text Generation • Updated Aug 23 • 39.3k • 3
ramsrigouthamg/t5_sentence_paraphraser
Text2Text Generation • Updated Jun 23 • 27.5k • 1
Michau/t5-base-en-generate-headline
Text2Text Generation • Updated Jun 22 • 35k • 8
ExT5
CodeT5
ProtT5
mT5
T5
T5.1.1
ByT5
UnifiedQA
UNICORN
T5+LM
MACAW
T0
Additional training
New model

php

GCC

7ZIP

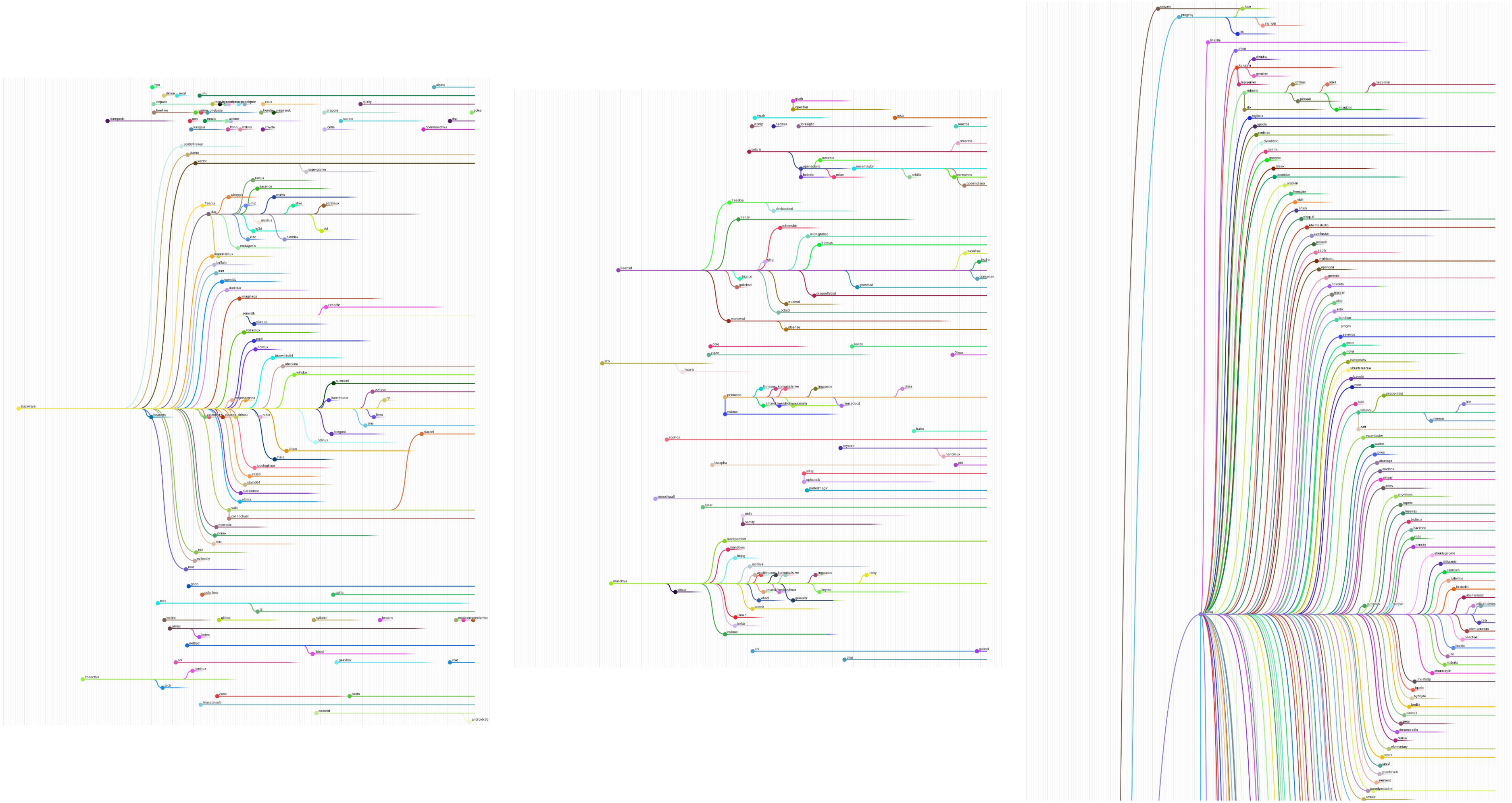
MySQL®

RAILS

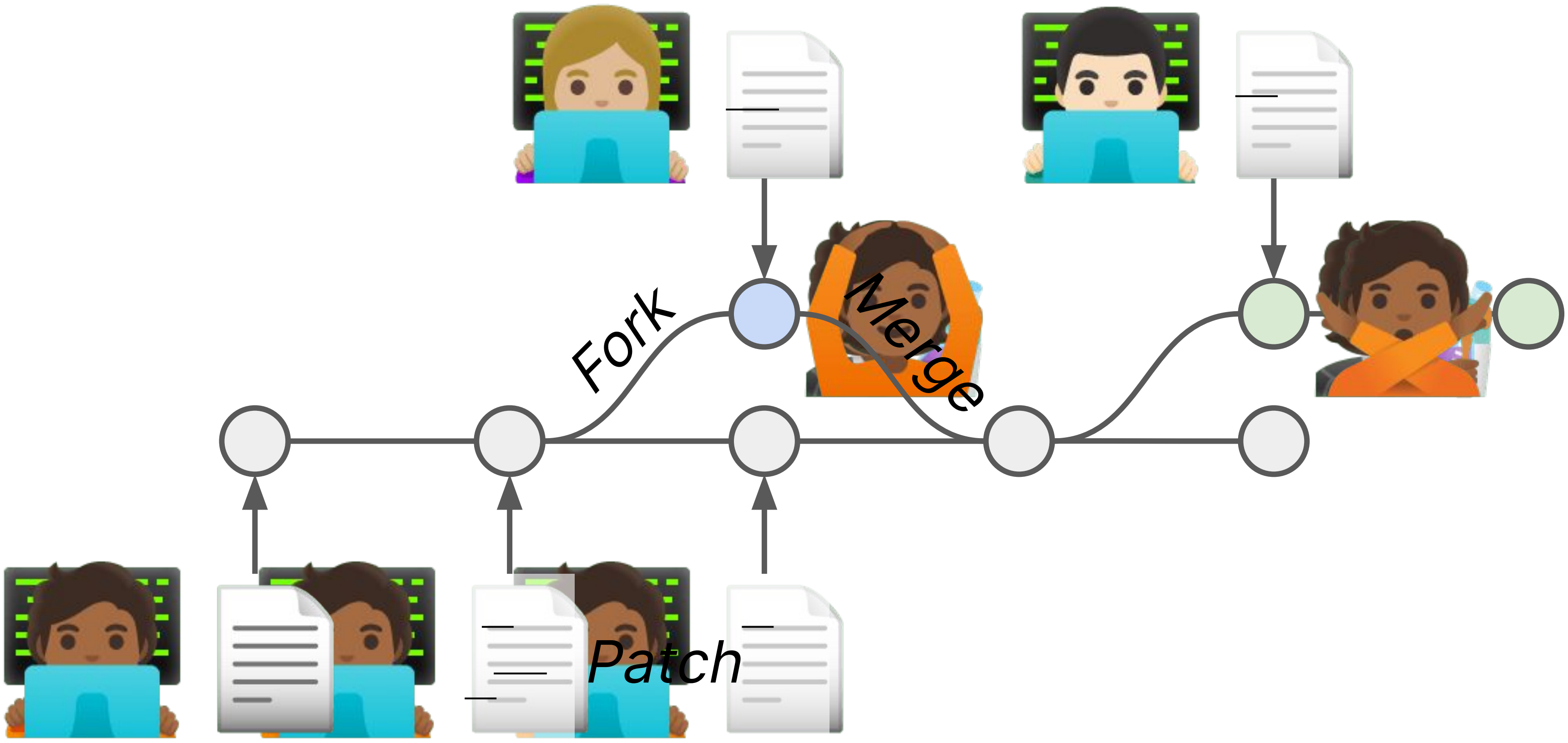
Open SSL

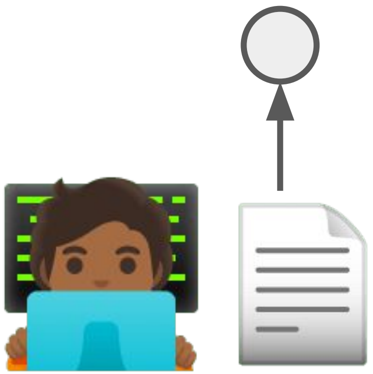
TOP SECRET

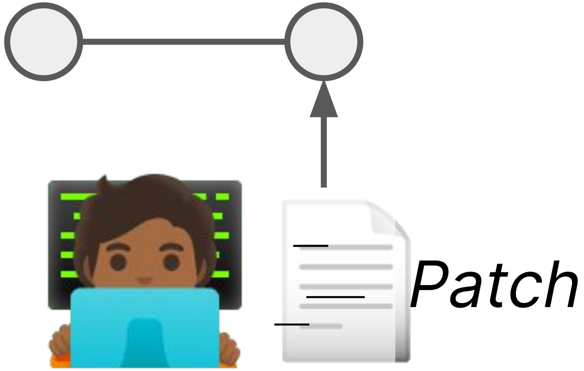
R

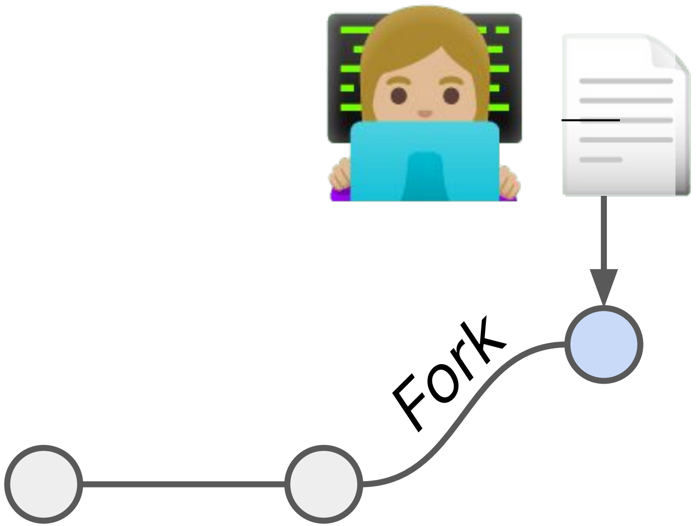


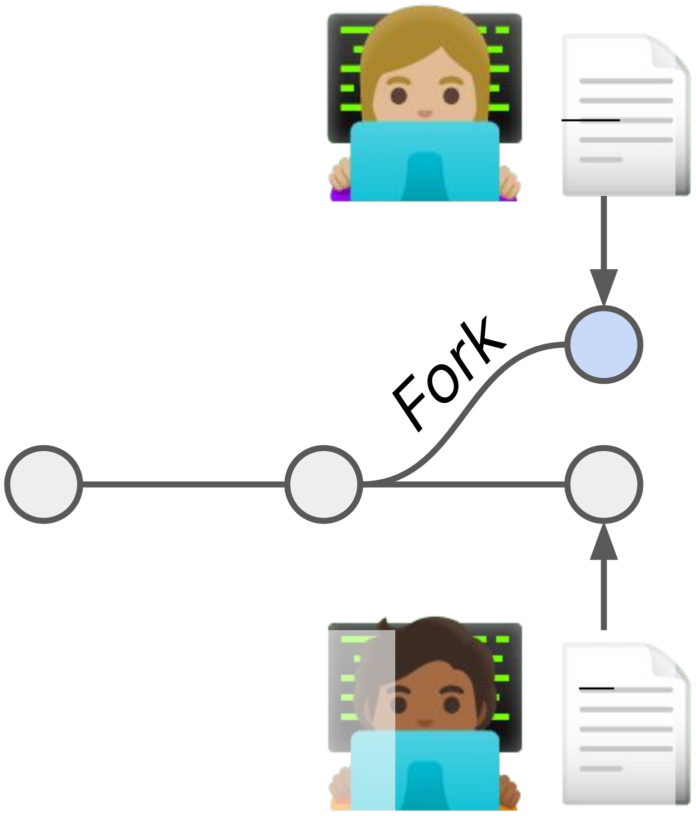
from https://distrowatch.com/

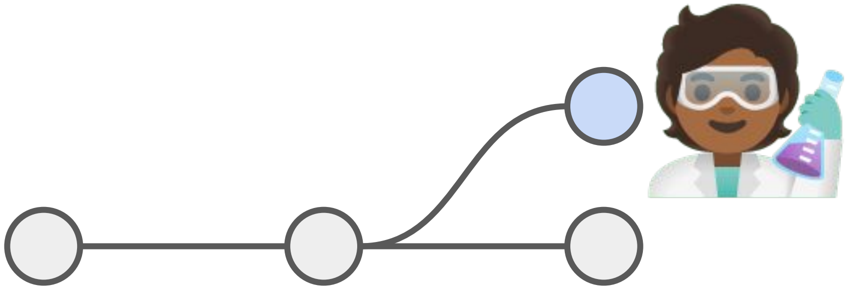

Fork
Merge
Patch

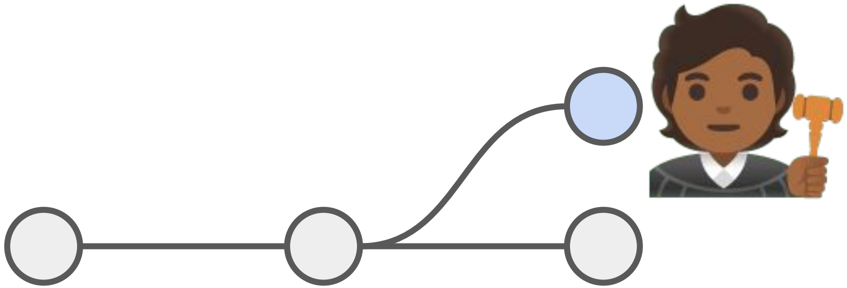


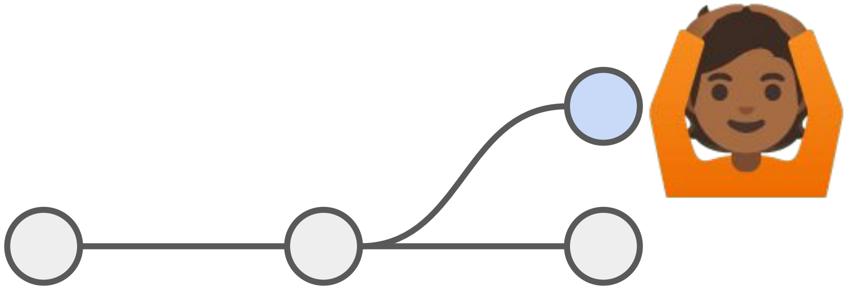

Patch

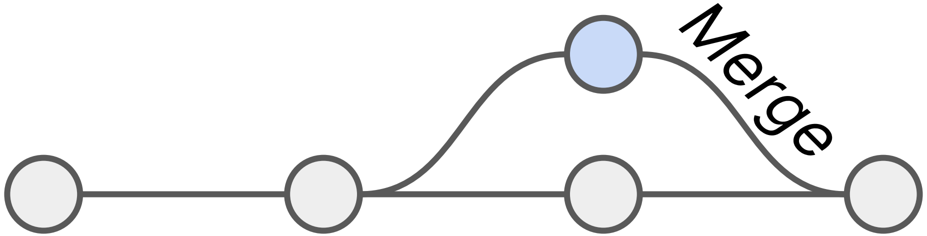

Fork

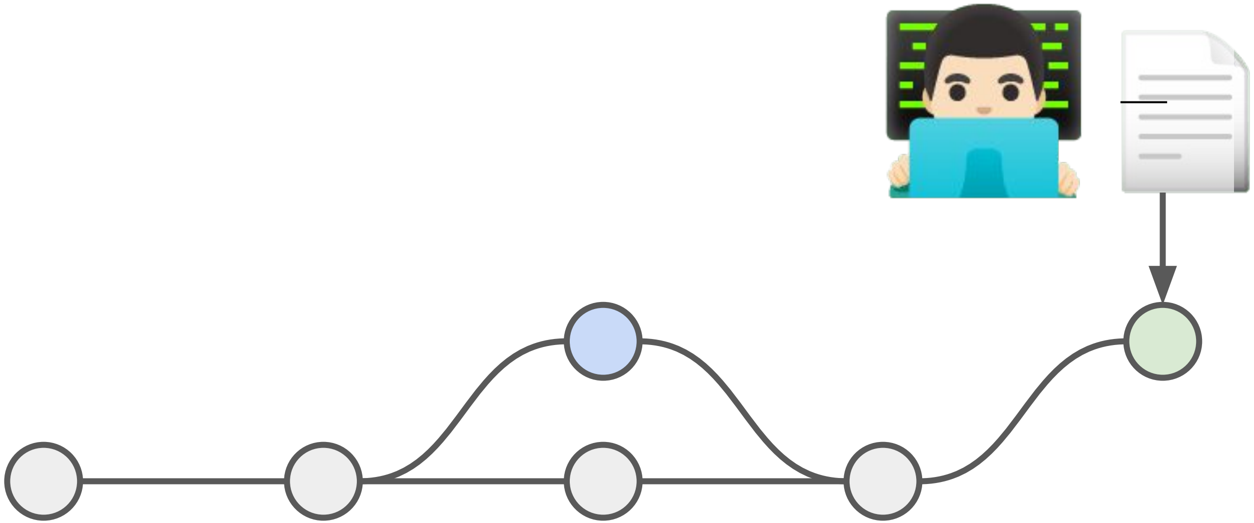

Fork

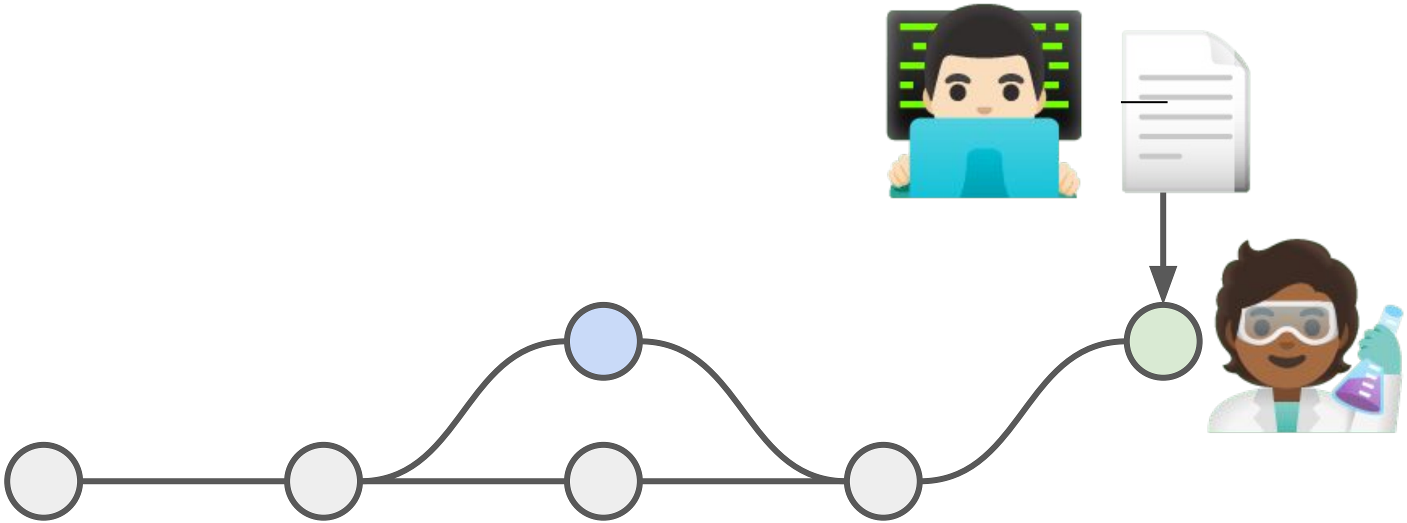


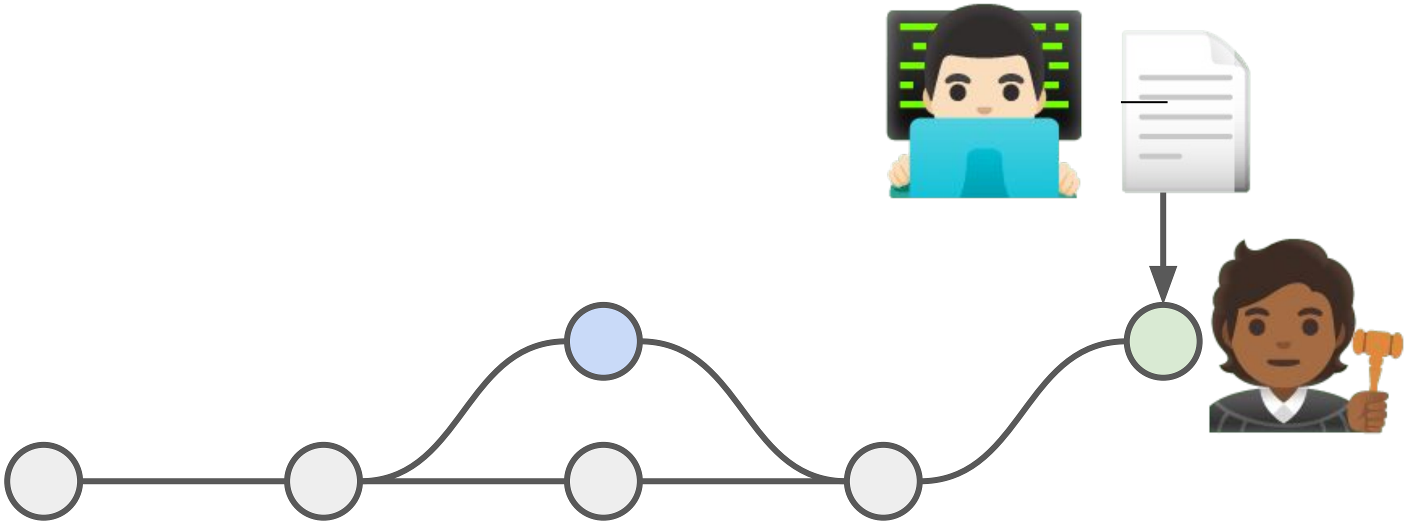


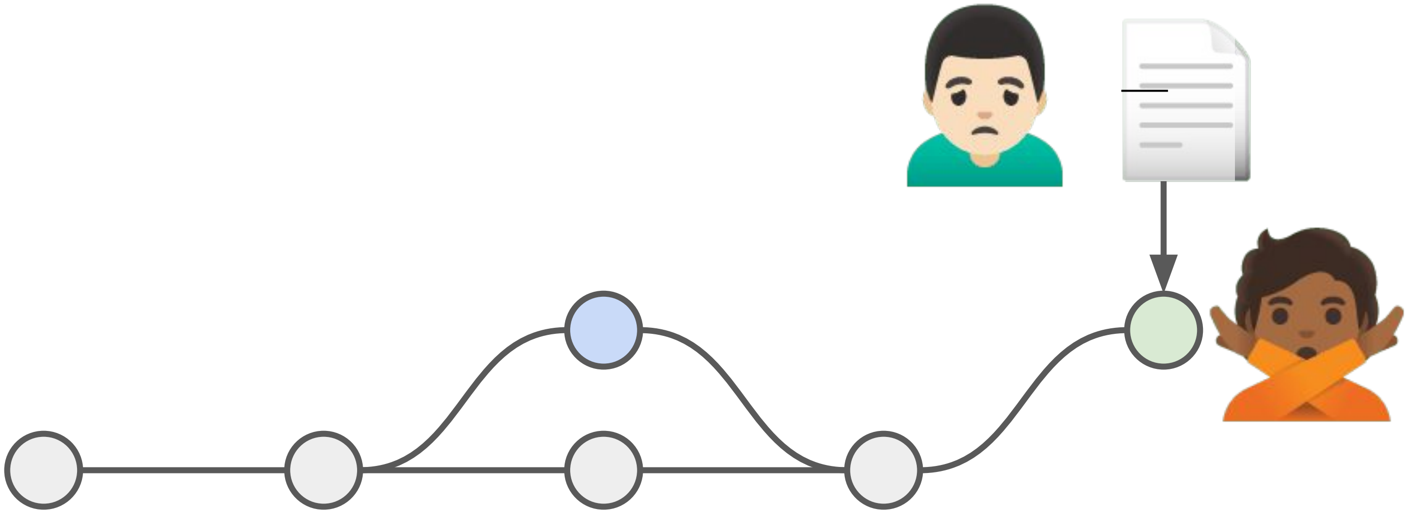


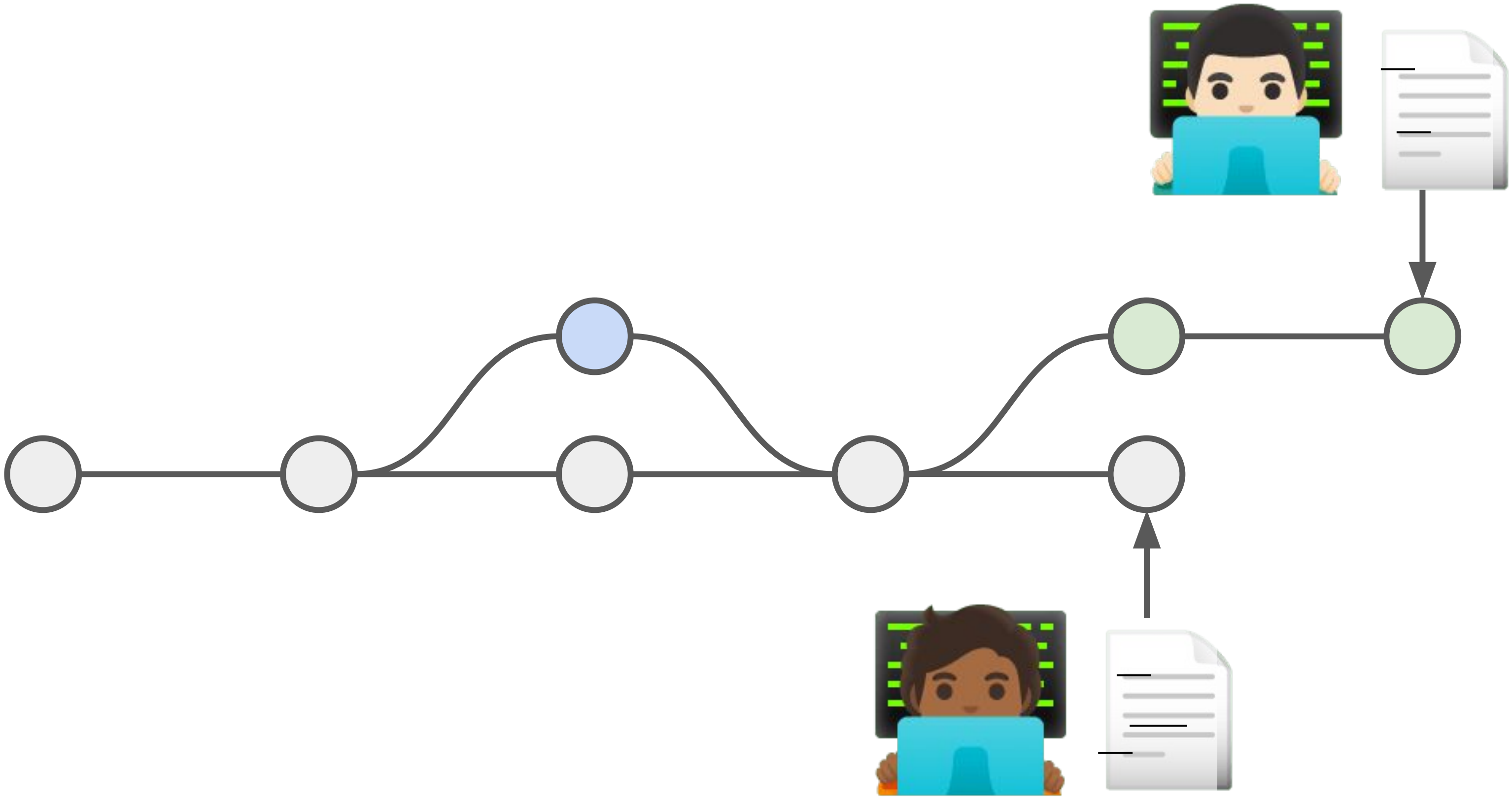

Merge

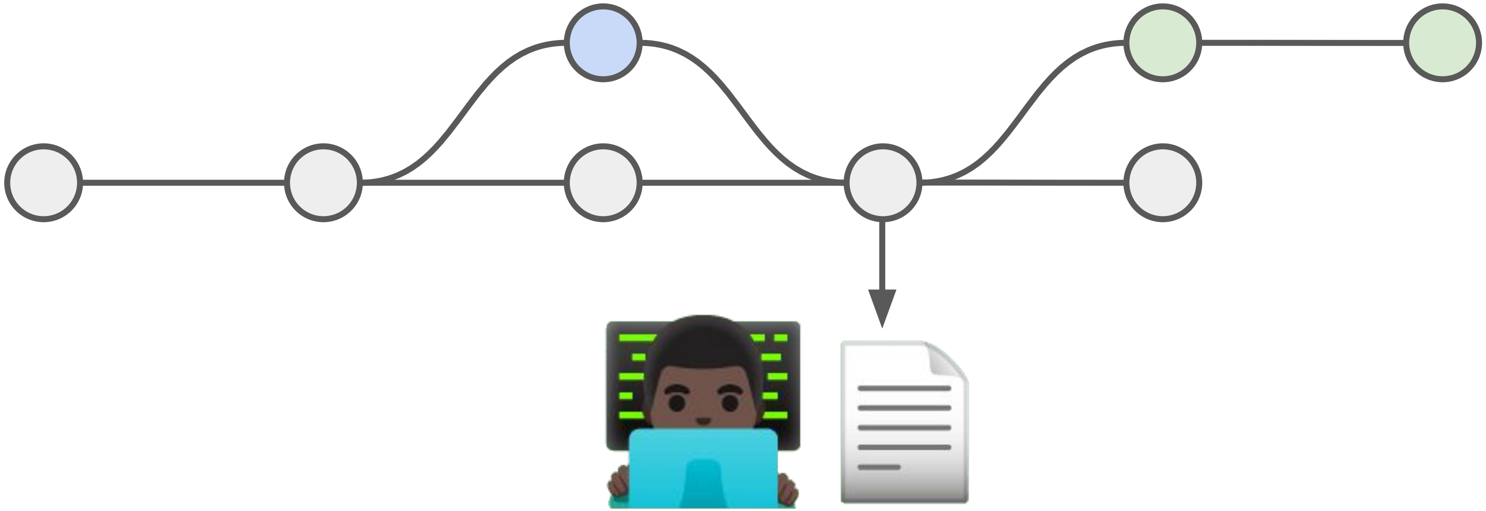












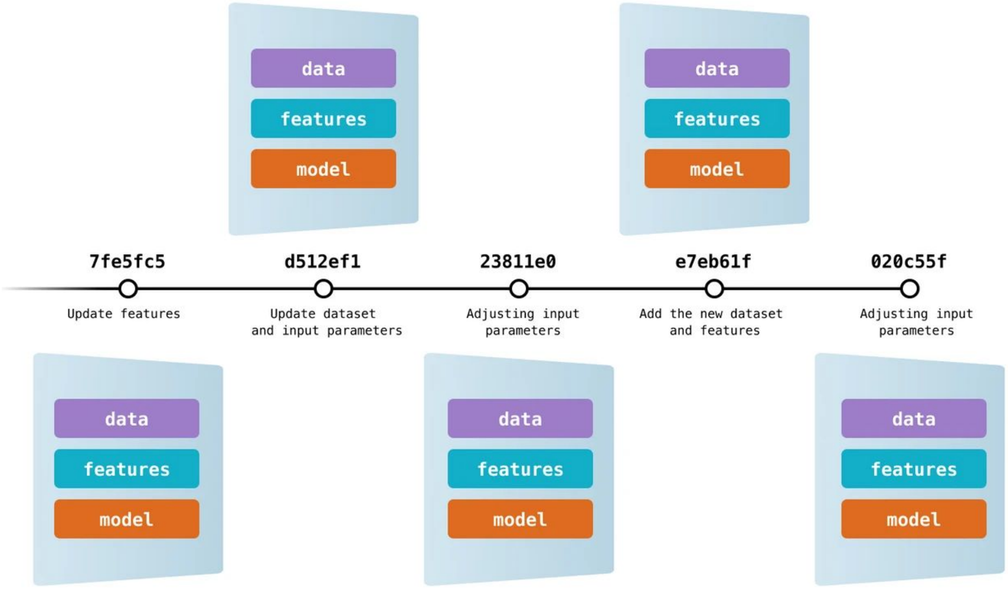
mlflow™

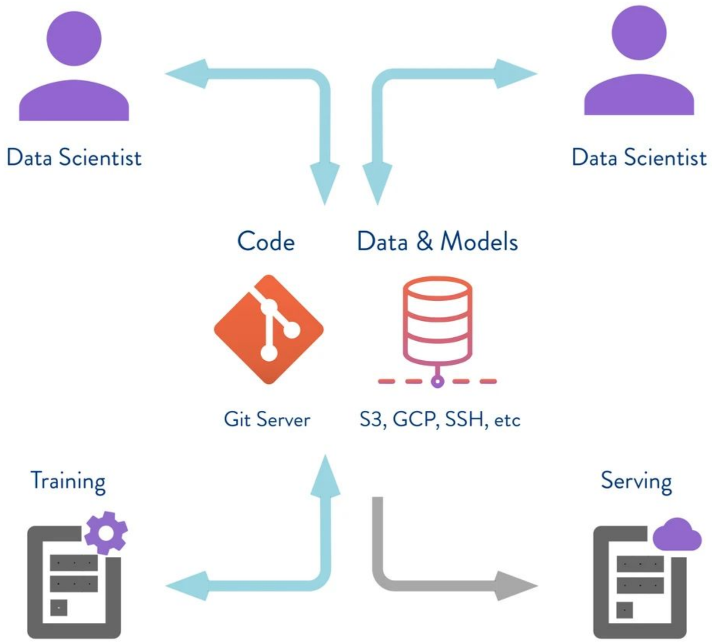
DVC

W&B

HUGGING FACE

comet





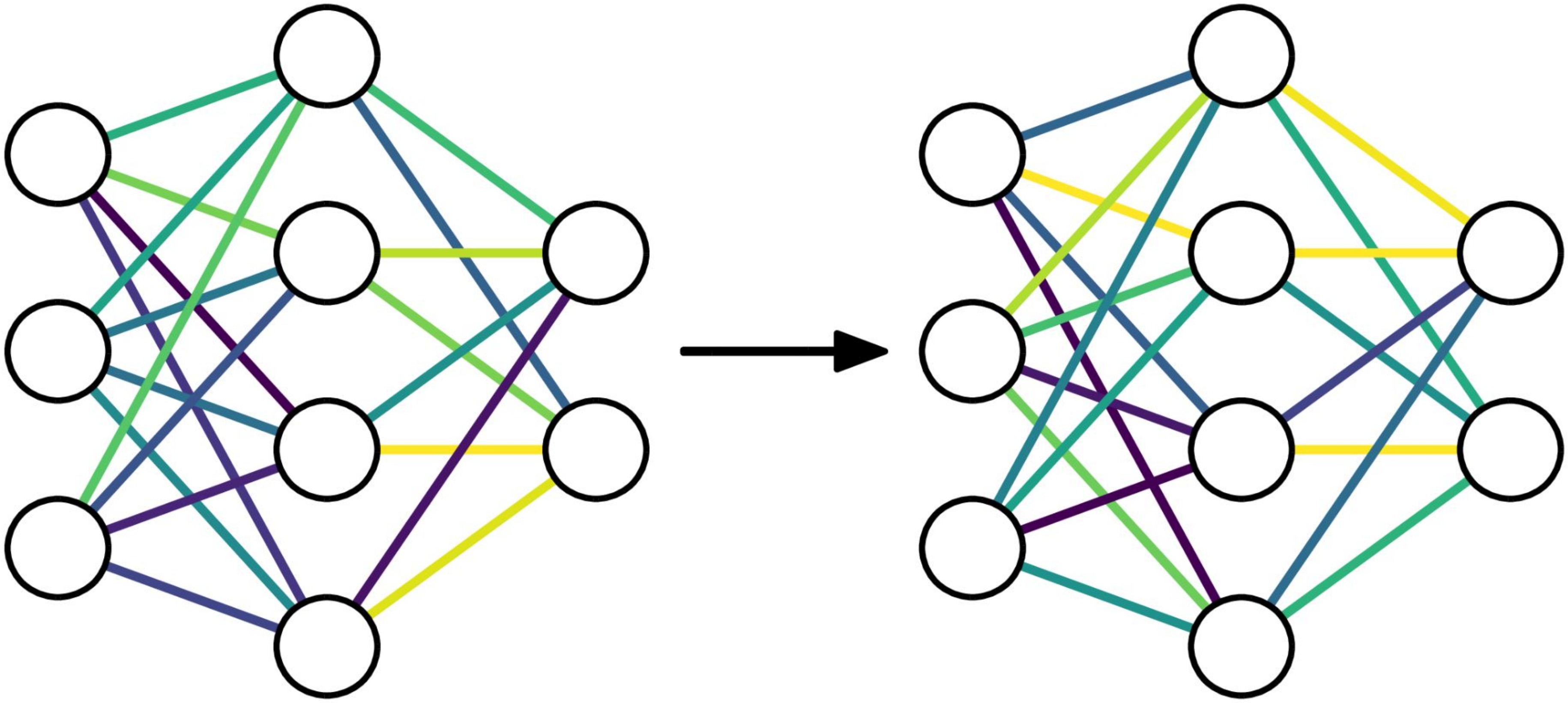
from https://dvc.org/

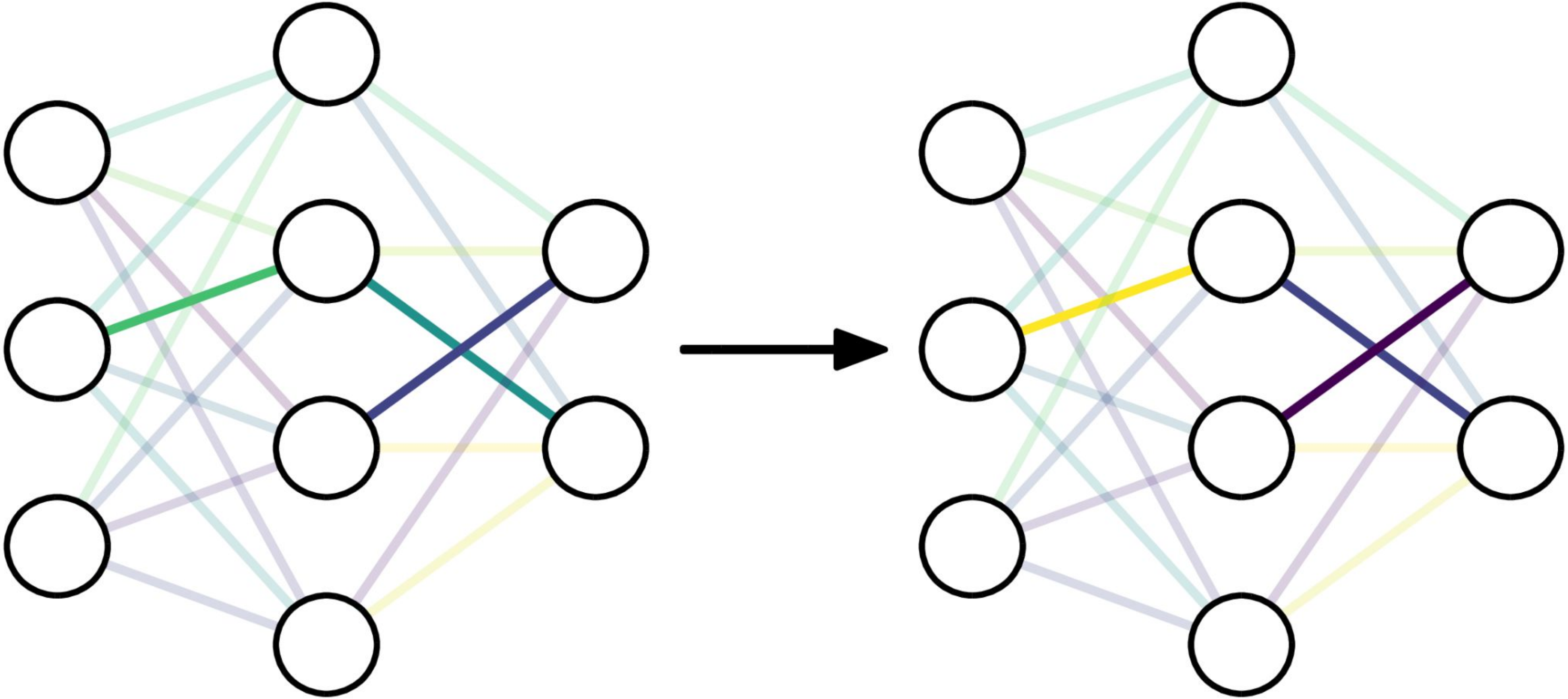
How can we enable collaborative and continual development of machine learning models?

We need to be able to cheaply communicate **patches** and **merge** updates from different contributors.

How can we enable collaborative and continual development of machine learning models?

We need to be able to cheaply communicate **patches** and **merge** updates from different contributors.





$$\mathrm{D_{KL}}(p_\theta(y|x) \mid\mid p_{\theta+\delta}(y|x))$$

$$\mathrm{D}_{\mathrm{KL}}(p_\theta(y|x) \mid\mid p_{\theta+\delta}(y|x))$$

$$\mathbb{E}_x \mathrm{D}_{\mathrm{KL}}(p_\theta(y|x) \mid\mid p_{\theta+\delta}(y|x)) = \delta^{\mathrm{T}} F_\theta \delta + O(\delta^3)$$

$$\mathrm{D_{KL}}(p_\theta(y|x) \,||\, p_{\theta+\delta}(y|x))$$

$$\mathbb{E}_x \mathrm{D_{KL}}(p_\theta(y|x) \,||\, p_{\theta+\delta}(y|x)) = \delta^{\mathrm{T}} F_\theta \delta + O(\delta^3)$$

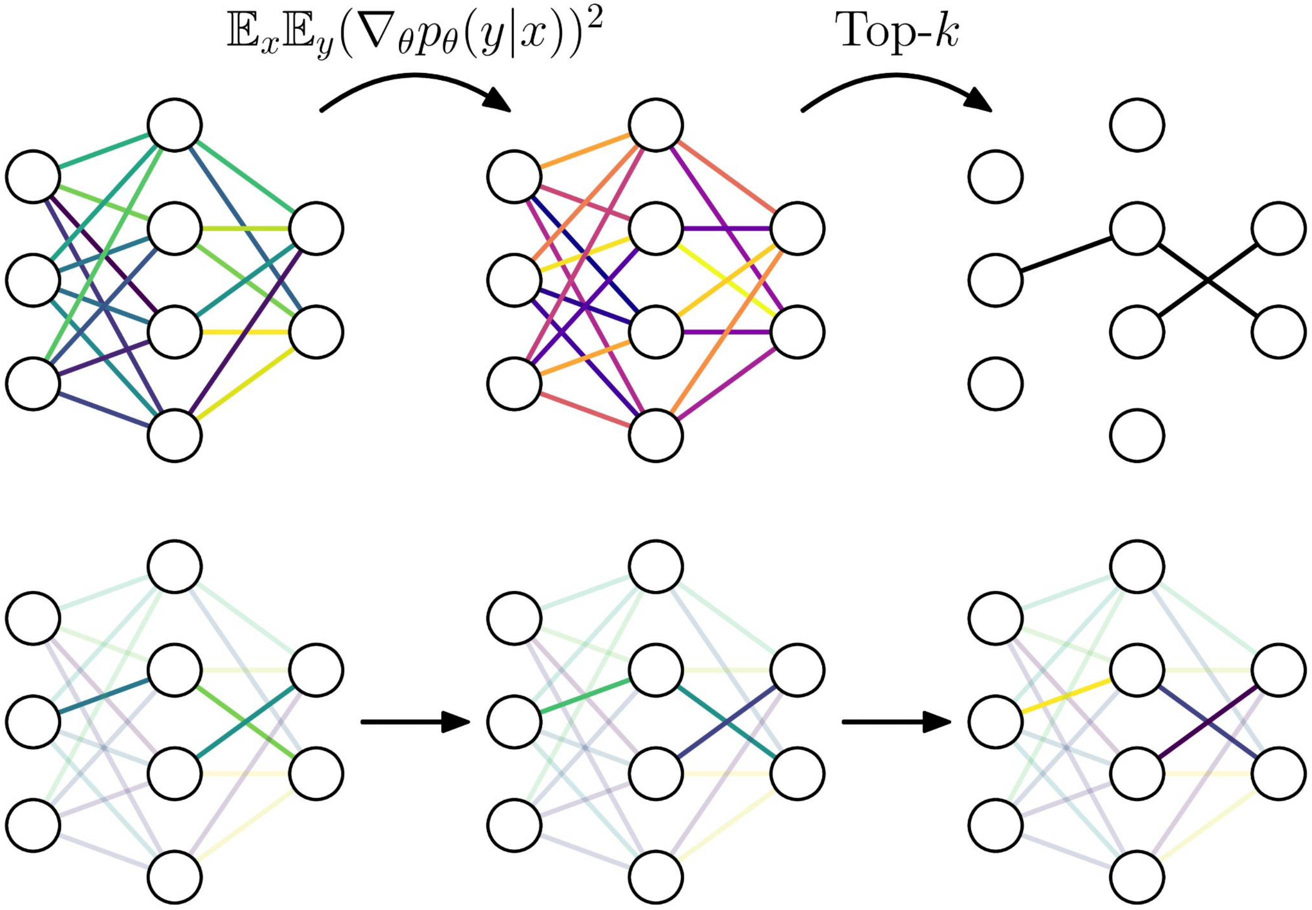
$$F_\theta = \mathbb{E}_{x\sim p(x)} \left[\mathbb{E}_{y\sim p_\theta(y|x)} \nabla_\theta \log p_\theta(y|x) \nabla_\theta \log p_\theta(y|x)^{\mathrm{T}}\right]$$

$$\mathrm{D}_{\mathrm{KL}}(p_\theta(y|x) \,||\, p_{\theta+\delta}(y|x))$$

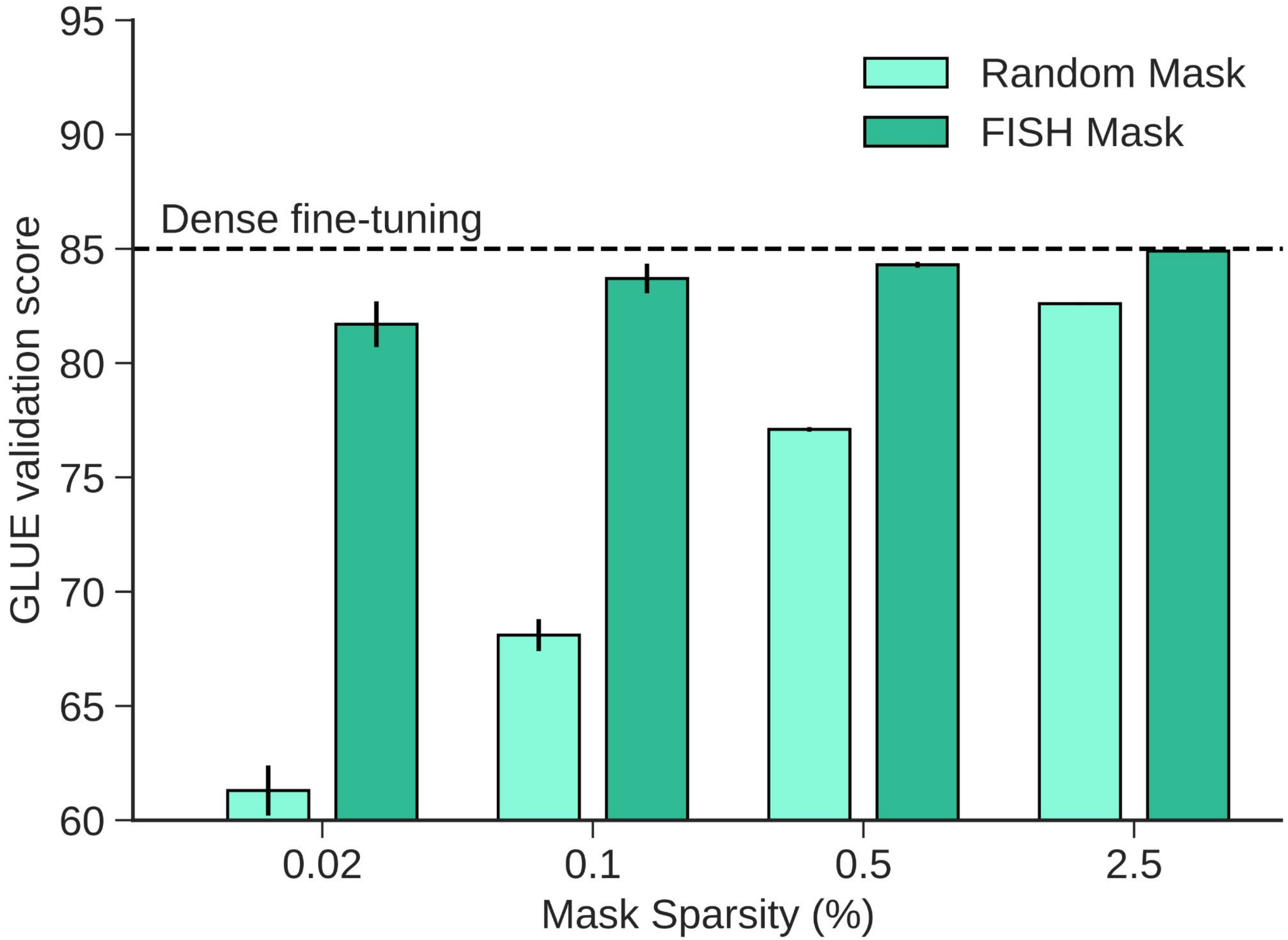
$$\mathbb{E}_x \mathrm{D}_{\mathrm{KL}}(p_\theta(y|x) \,||\, p_{\theta+\delta}(y|x)) = \delta^{\mathrm{T}} F_\theta \delta + O(\delta^3)$$

$$F_\theta = \mathbb{E}_{x\sim p(x)} \left[\mathbb{E}_{y\sim p_\theta(y|x)} \nabla_\theta \log p_\theta(y|x) \nabla_\theta \log p_\theta(y|x)^{\mathrm{T}} \right]$$

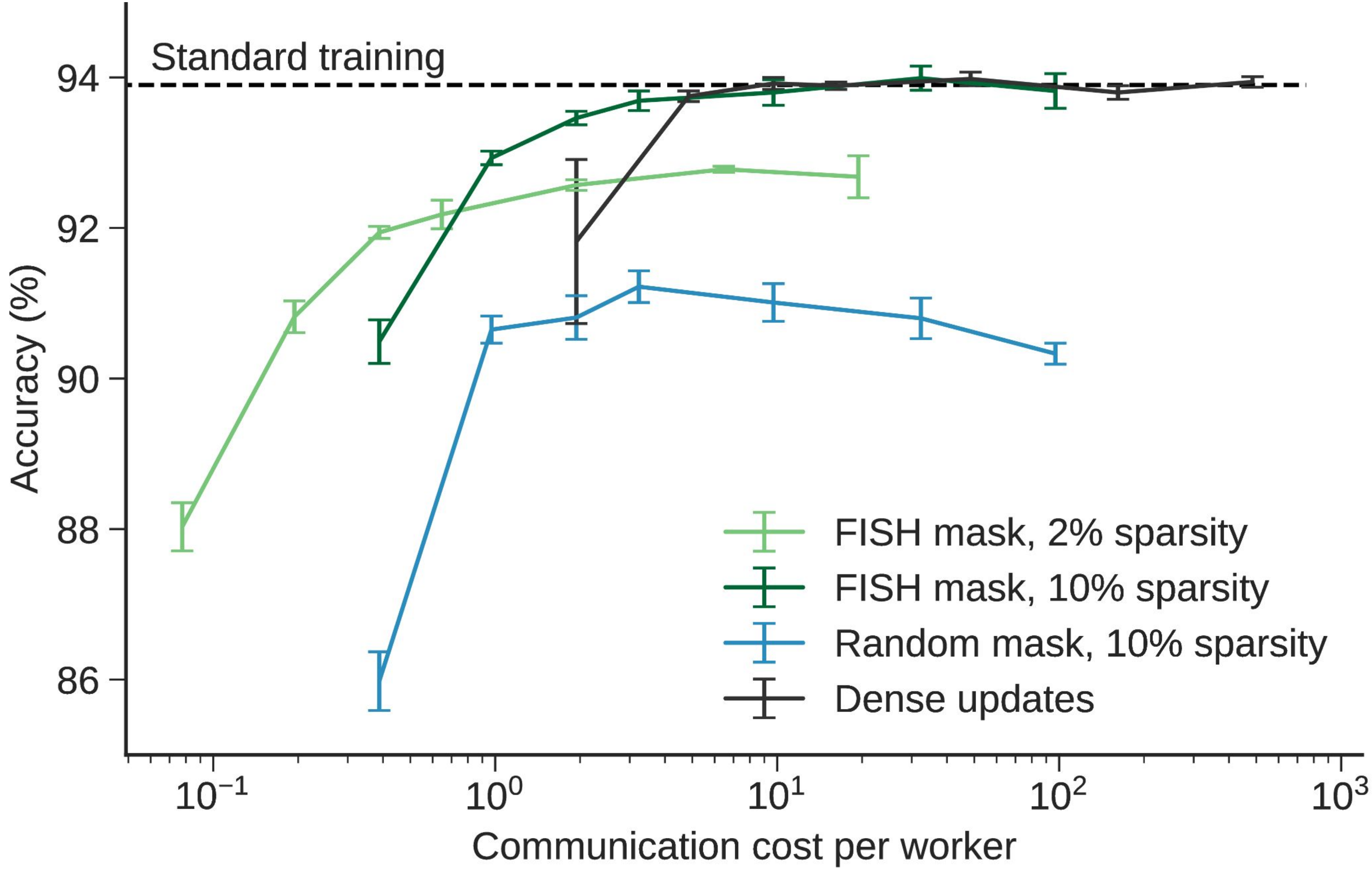
$$\hat{F}_\theta = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{y\sim p_\theta(y|x_i)} (\nabla_\theta \log p_\theta(y|x_i))^2$$



***F**isher-**I**nduced **S**parse Unc**h**anging (FISH) Mask*


95
90
85
80
75
70
65
60
GLUE validation score
Dense fine-tuning
Random Mask
FISH Mask
0.02
0.1
0.5
2.5
Mask Sparsity (%)

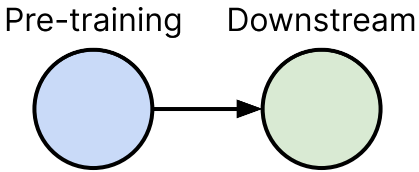
Method	Sparsity	GLUE Score
Dense Fine-tuning	100%	82.5
Bit-Fit	0.08%	81.2
FISH Mask	0.08%	81.3
Diff Pruning	0.50%	81.5
FISH Mask	0.50%	82.6

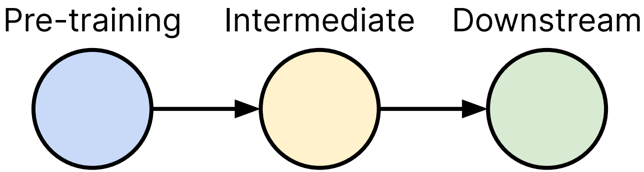


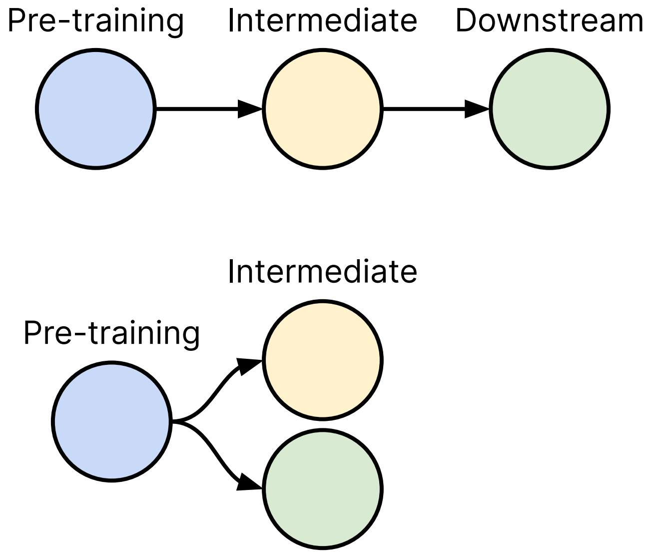
Standard training
94
92
90
88
86
Accuracy (%)
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Communication cost per worker
FISH mask, 2% sparsity
FISH mask, 10% sparsity
Random mask, 10% sparsity
Dense updates

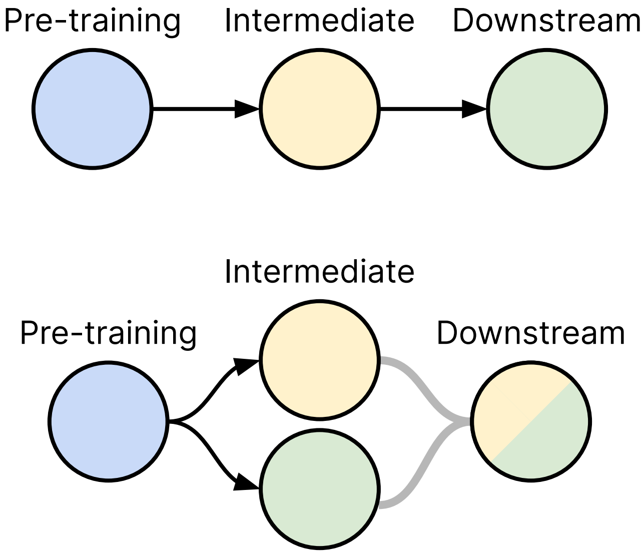
How can we enable collaborative and continual development of machine learning models?

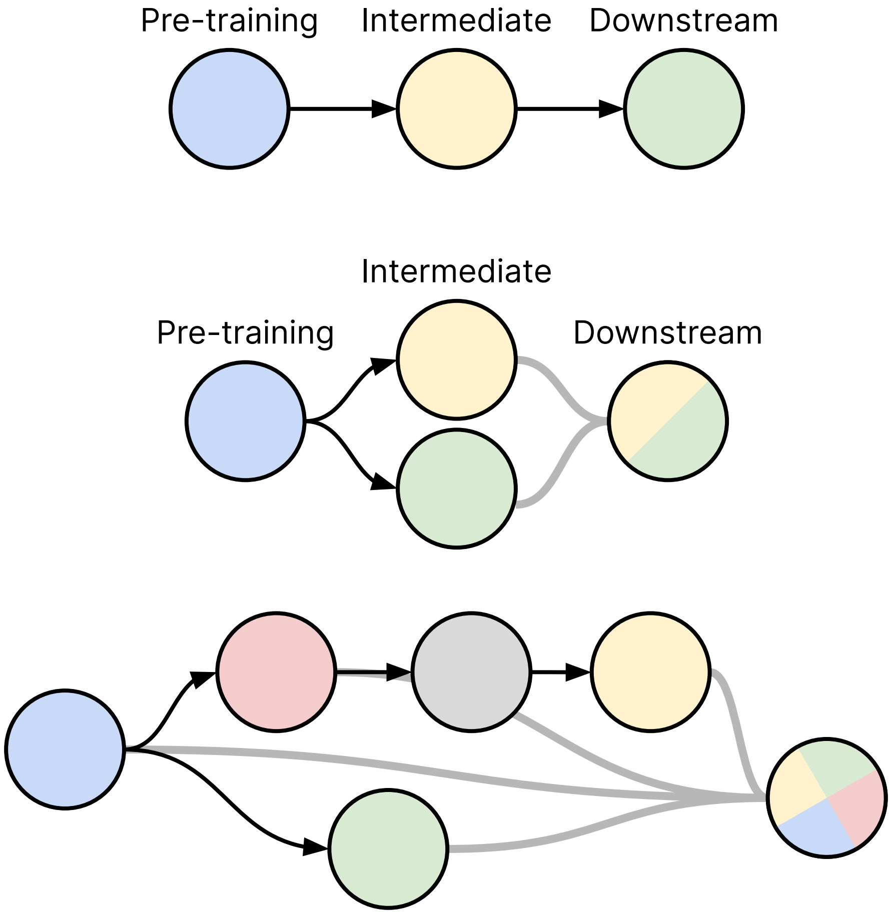
We need to be able to cheaply communicate **patches** and **merge** updates from different contributors.


Pre-training
Downstream


Pre-training
Intermediate
Downstream


Pre-training
Intermediate
Downstream
Intermediate
Pre-training


Pre-training
Intermediate
Downstream
Intermediate
Pre-training
Downstream


Pre-training
Intermediate
Downstream
Intermediate
Pre-training
Downstream

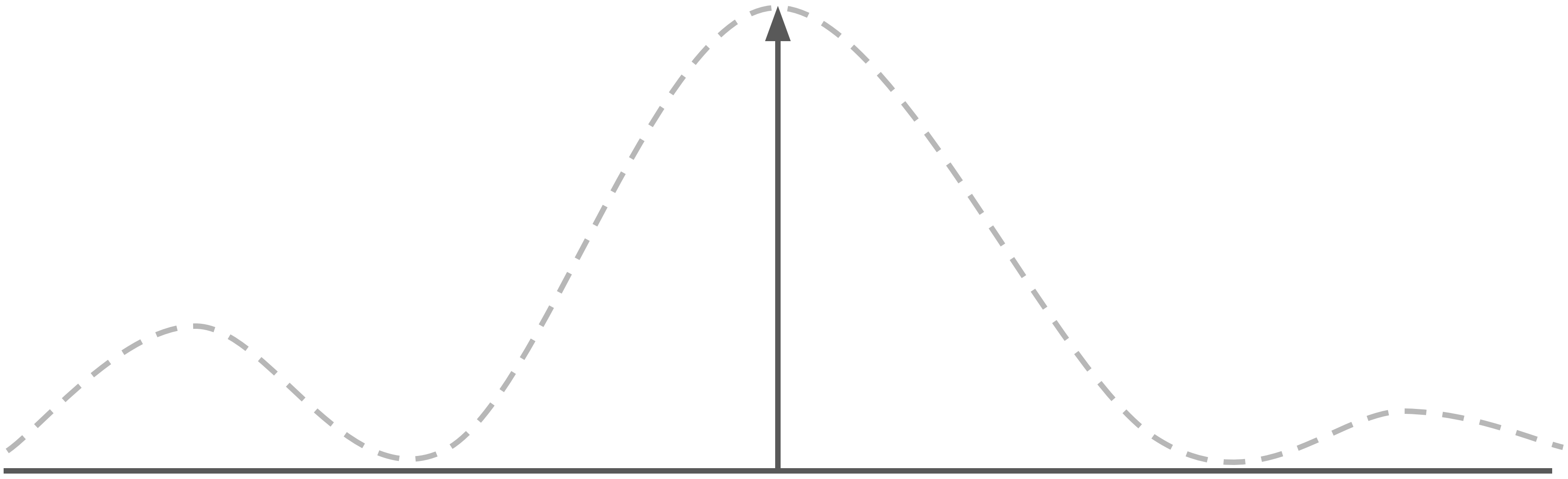
$$\arg\max_{\theta} \sum_{i=1}^{M} \lambda_i \log p(\theta | \mathcal{D}_i)$$

$$\arg\max_{\theta} \sum_{i=1}^{M} \lambda_i \underbrace{\log p(\theta|\mathcal{D}_i)}_{\textit{Log posterior for model i}}$$

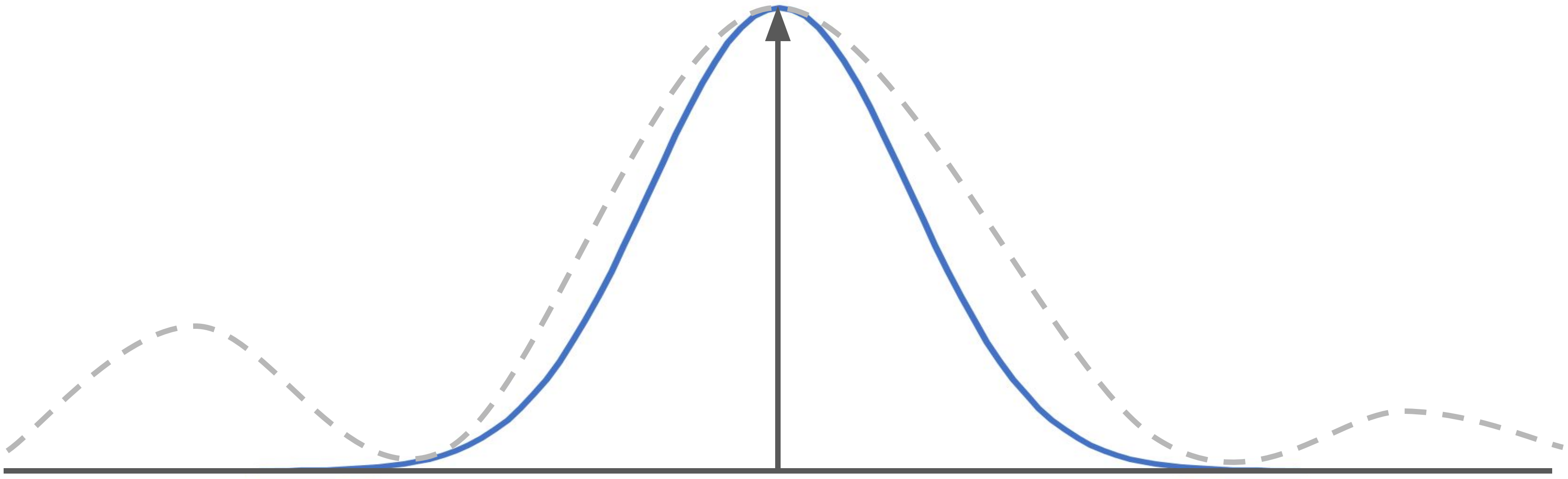
$$\arg\max_{\theta} \sum_{i=1}^{M} \lambda_i \log p(\theta|\mathcal{D}_i)$$

Hyperparameter controlling the importance of model i

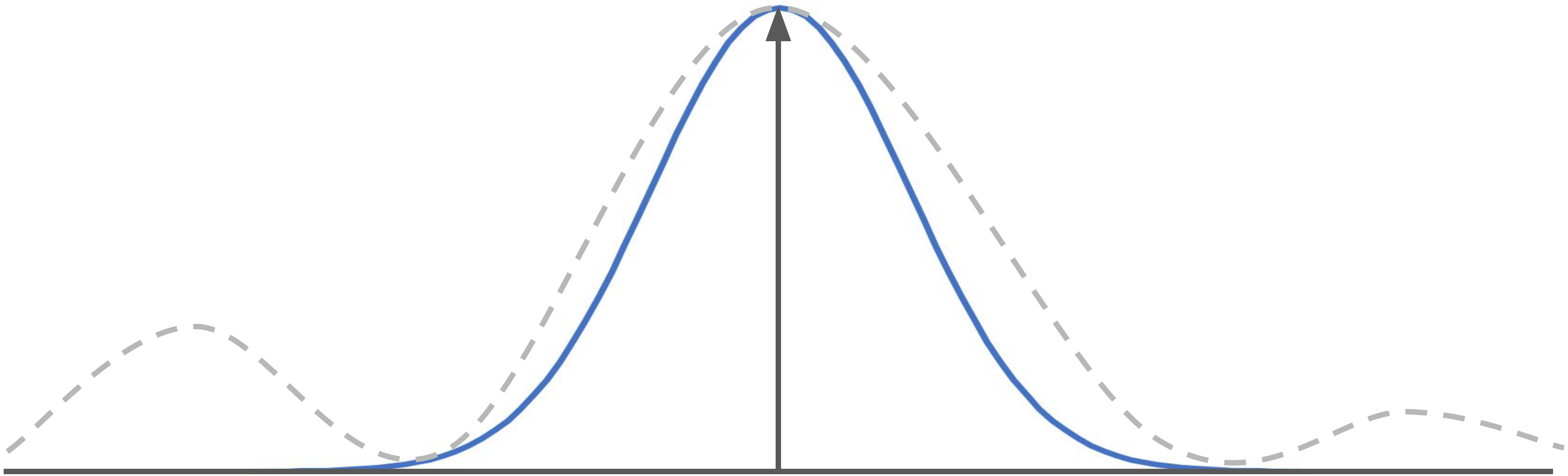
$$\arg\max_{\theta} \sum_{i=1}^{M} \lambda_i \log p(\quad i)$$



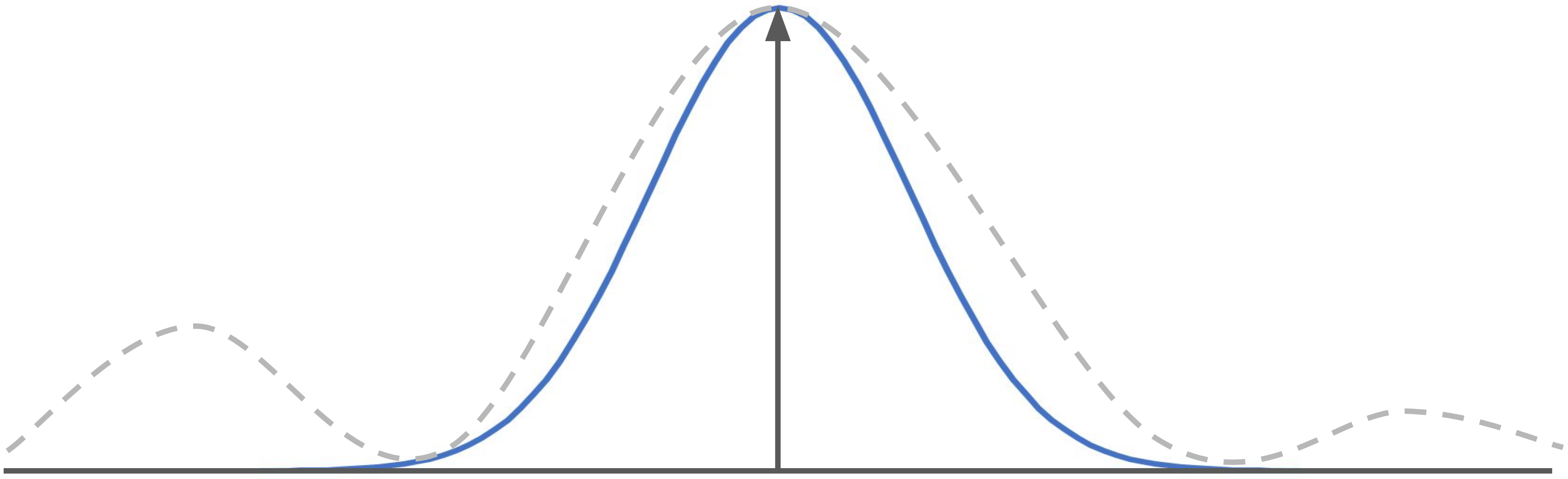
$$\arg\max_{\theta} \sum_{i=1}^{M} \lambda_i \log \mathcal{N}(\theta | \theta_i, H_i^{-1})$$



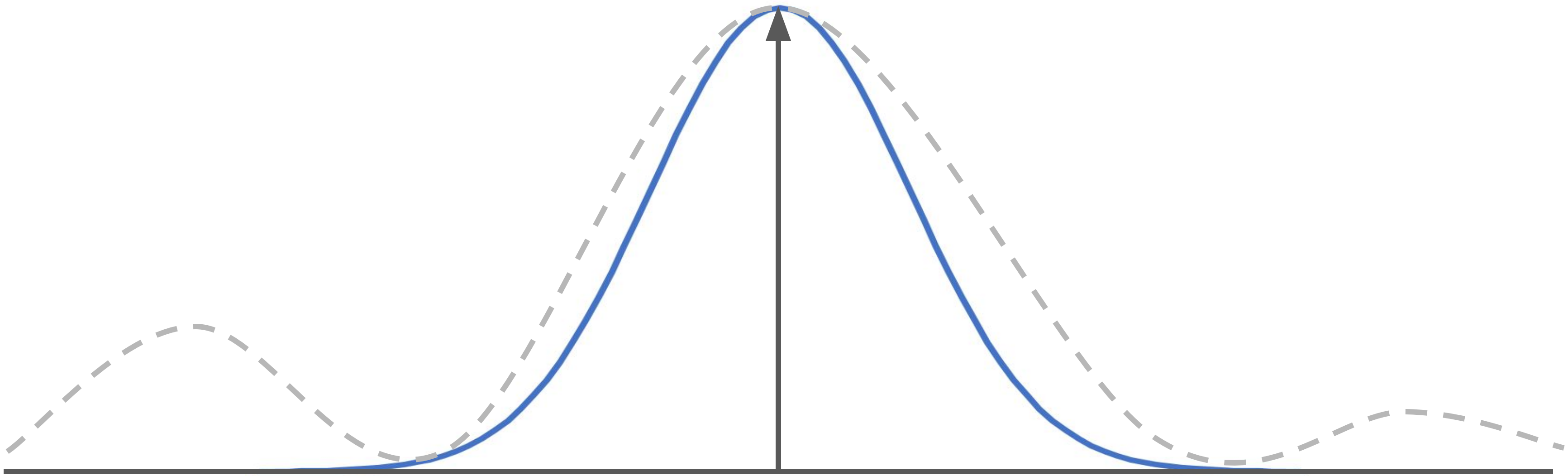
$$\arg\max_{\theta} \sum_{i=1}^{M} \lambda_i \log \mathcal{N}(\theta|\theta_i, \quad)$$



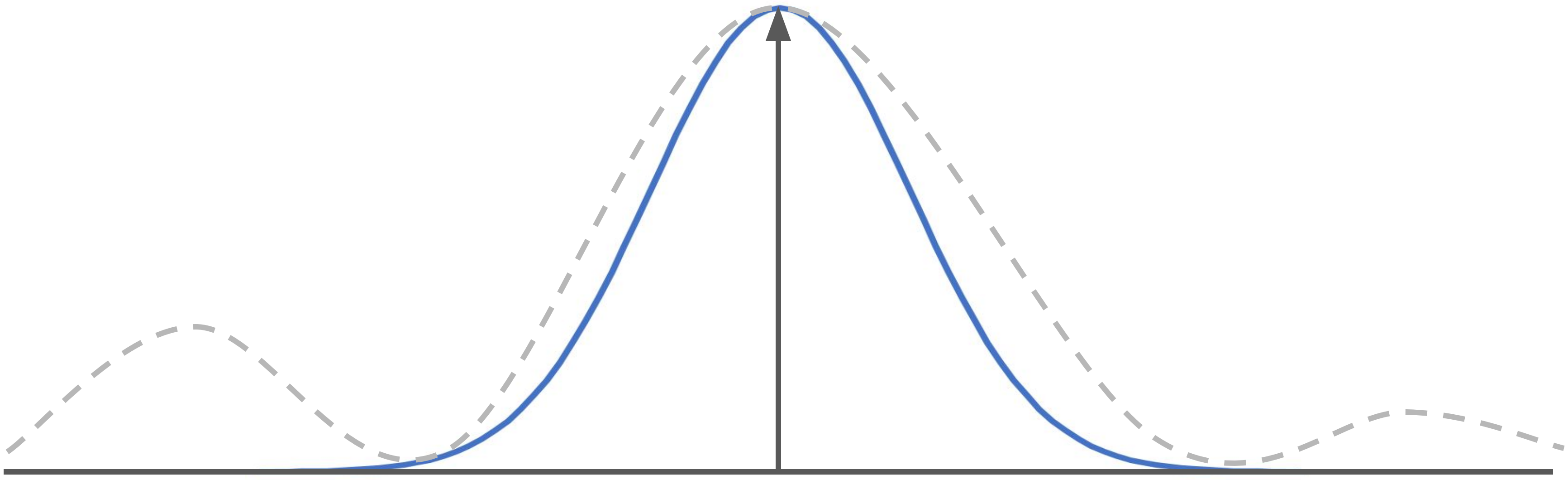
$$\arg\max_{\theta} \sum_{i=1}^{M} \lambda_i \log \mathcal{N}(\theta | \theta_i, F_i)$$



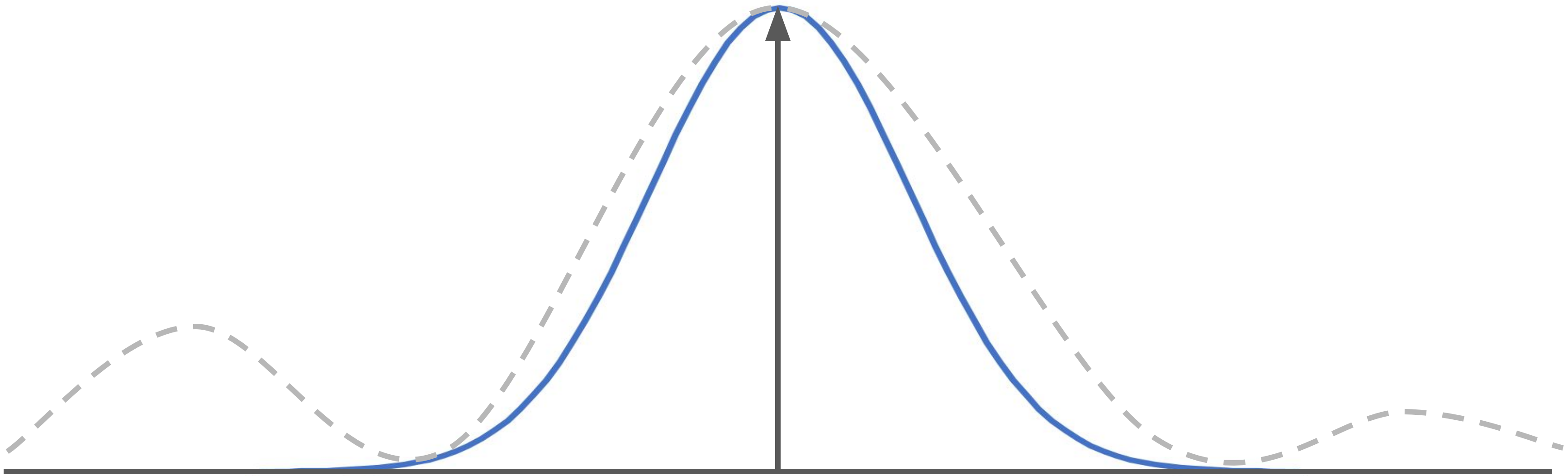
$$\arg\max_{\theta}\sum_{i=1}^{M}\lambda_i \log \mathcal{N}(\theta|\theta_i, \;)$$

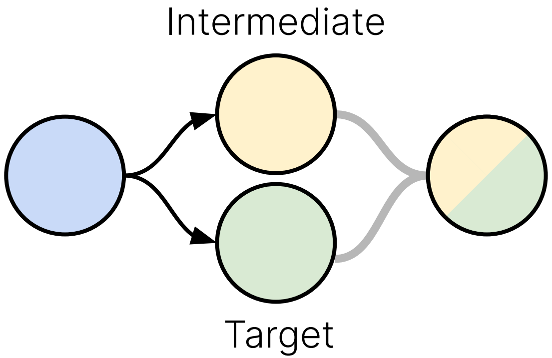


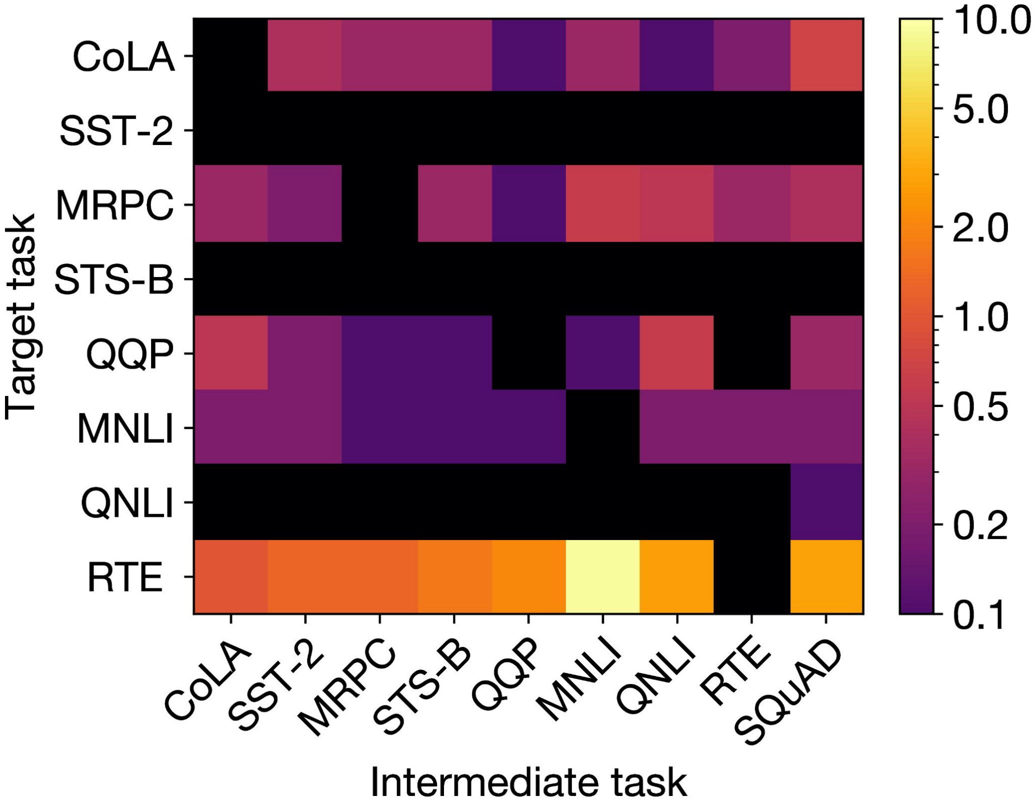
$$\arg\max_{\theta} \sum_{i=1}^{M} \lambda_i \log \mathcal{N}(\theta | \theta_i, \hat{F}_i)$$

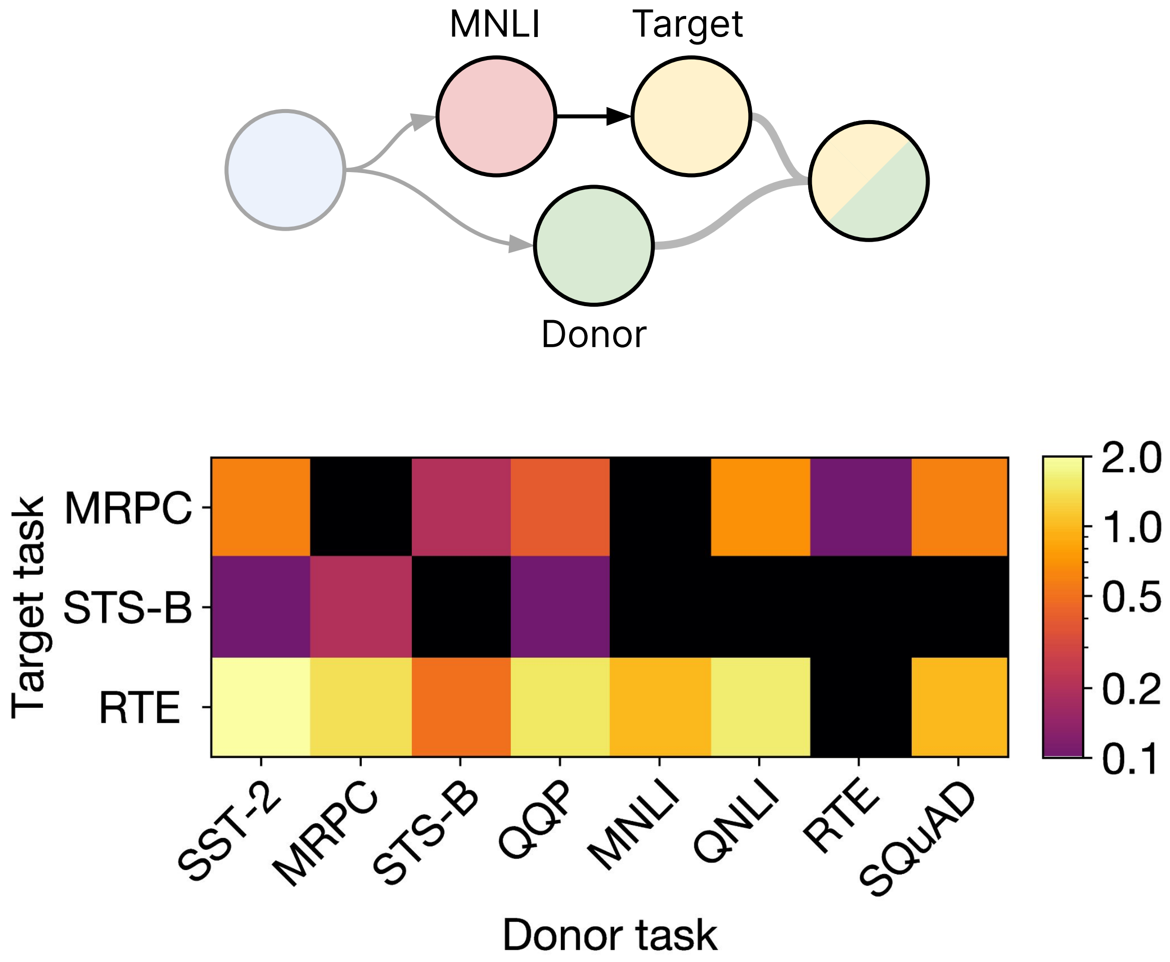


$$\arg\max_{\theta} \sum_{i=1}^{M} \lambda_i \log \mathcal{N}(\theta | \theta_i, \hat{F}_i)$$

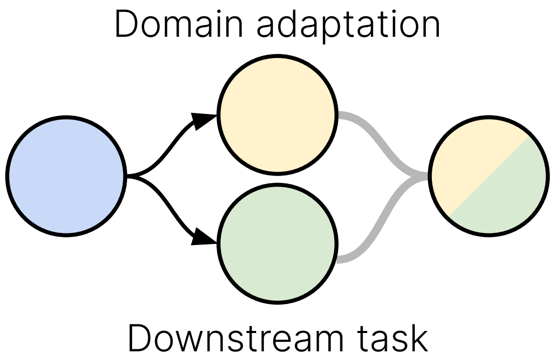



Intermediate
Target


Target task
CoLA
SST-2
MRPC
STS-B
QQP
MNLI
QNLI
RTE
Intermediate task
CoLA
SST-2
MRPC
STS-B
QQP
MNLI
QNLI
RTE
SQuAD
10.0
5.0
2.0
1.0
0.5
0.2
0.1



MNLI
Target
Donor
MRPC
STS-B
RTE
Target task
2.0
1.0
0.5
0.2
0.1
SST-2
MRPC
STS-B
QQP
MNLI
QNLI
RTE
SQuAD
Donor task



Task	Unmerged	Merged	Fine-tuned
ChemProt	$82.7_{0.3}$	$83.1_{0.4}$	$82.5_{0.1}$
ACL-ARC	$70.5_{3.2}$	$73.2_{1.7}$	$71.5_{3.0}$
SciERC	$81.0_{0.4}$	$81.3_{0.5}$	$81.6_{1.0}$

How can we enable collaborative and continual development of machine learning models?

We need to be able to cheaply communicate **patches** and **merge** updates from different contributors.

How can we enable collaborative and continual development of machine learning models?

We need to be able to **rapidly evaluate** proposed changes to the model to ensure backward compatibility.

How can we enable collaborative and continual development of machine learning models?

We need to be able to combine **modular** components of different models to provide new skills and capabilities.

[Training Neural Networks with Fixed Sparse Masks]
Yi-Lin Sung, Varun Nair, and Colin Raffel

[Merging Models with Fisher-Weighted Averaging]
Michael Matena and Colin Raffel

Please give me feedback:
http://bit.ly/colin-talk-feedback