# 一种通用快速准确的预训练模型评估方法

清华大学 游凯超

2021.3.10

扫码关注，获取最新资讯

# 多种了解渠道

- 公众号推送（快速了解大致内容）

- Live报告（快速了解技术细节）
- 论文（完整了解全部内容）
  - https://arxiv.org/abs/2102.11005
- 代码（快速在项目中使用）
  - https://github.com/thuml/LogME

# 自我介绍

- 清华大学软件学院机器学习组
- 2020级博士研究生
- 师从 龙明盛老师
- 研究领域：迁移学习
- 个人主页 youkaichao.github.io
  - 知乎、谷歌学术、邮箱……

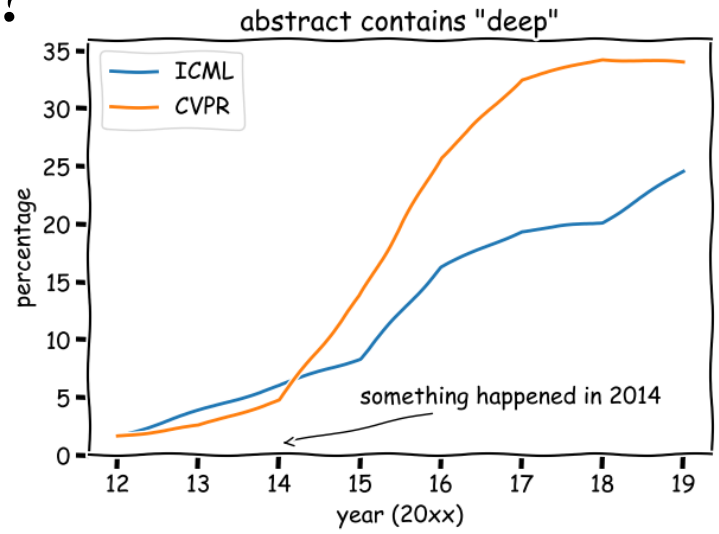
- ◀️2019智源live
  - 领域适配前沿研究--场景、方法与模型选择

# 目录

- 问题引入
  - 迁移学习范式
  - 预训练模型选择
- 现有方法
  - 暴力搜索
  - LEEP/NCE
- LogME
  - 推导过程
  - 算法优化
- 实验效果
- LogME的扩展及应用场景

# 迁移学习范式

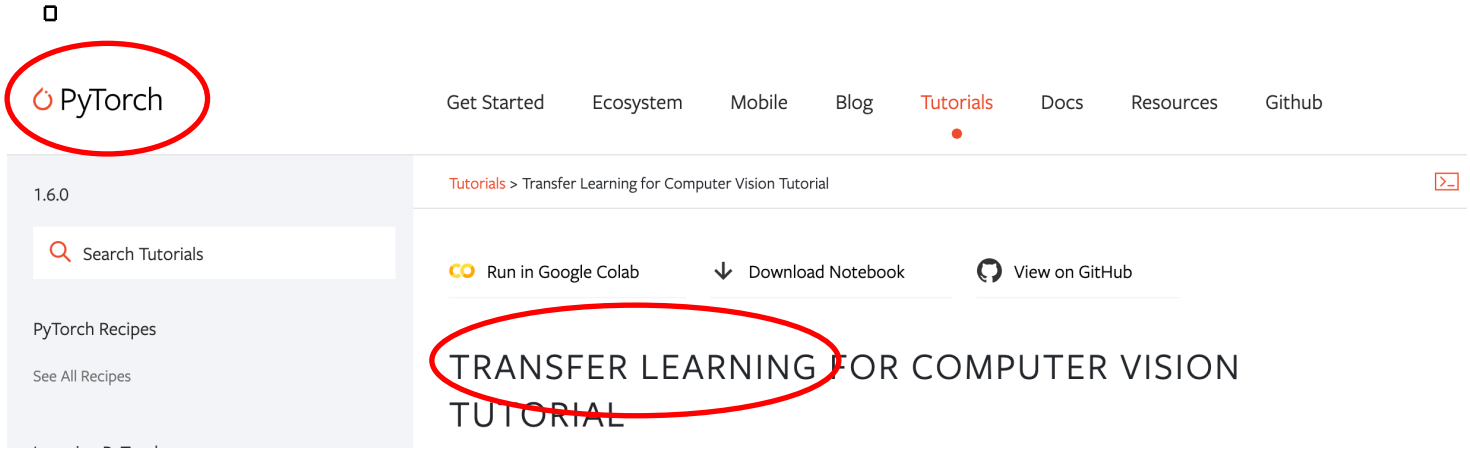
- AlexNet是深度学习流行的原因？
- 用数据说话
  - 顶会论文摘要中包含deep的比例
  - 2012年，很少有deep
  - 2012年开始增长
  - （很少有人发现）2014年的拐点

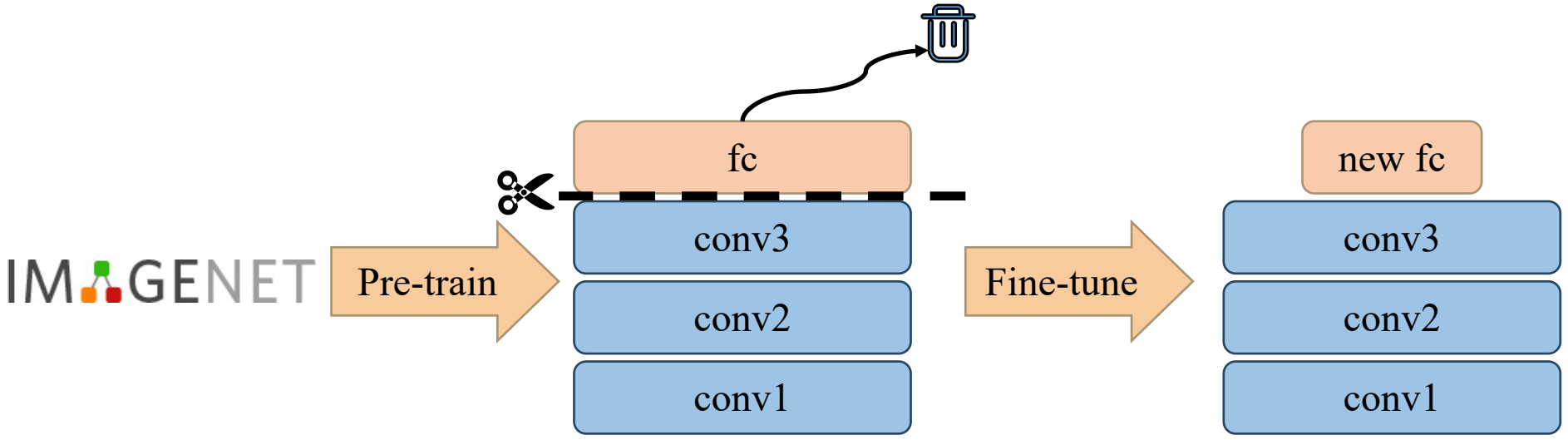



- 2014年的重磅论文
  - Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR 2014 (citation over 1.5w)
  - Decaf: A deep convolutional activation feature for generic visual recognition, ICML 2014 (citation over 4k)
  - 都是迁移学习

https://youkaichao.github.io/about

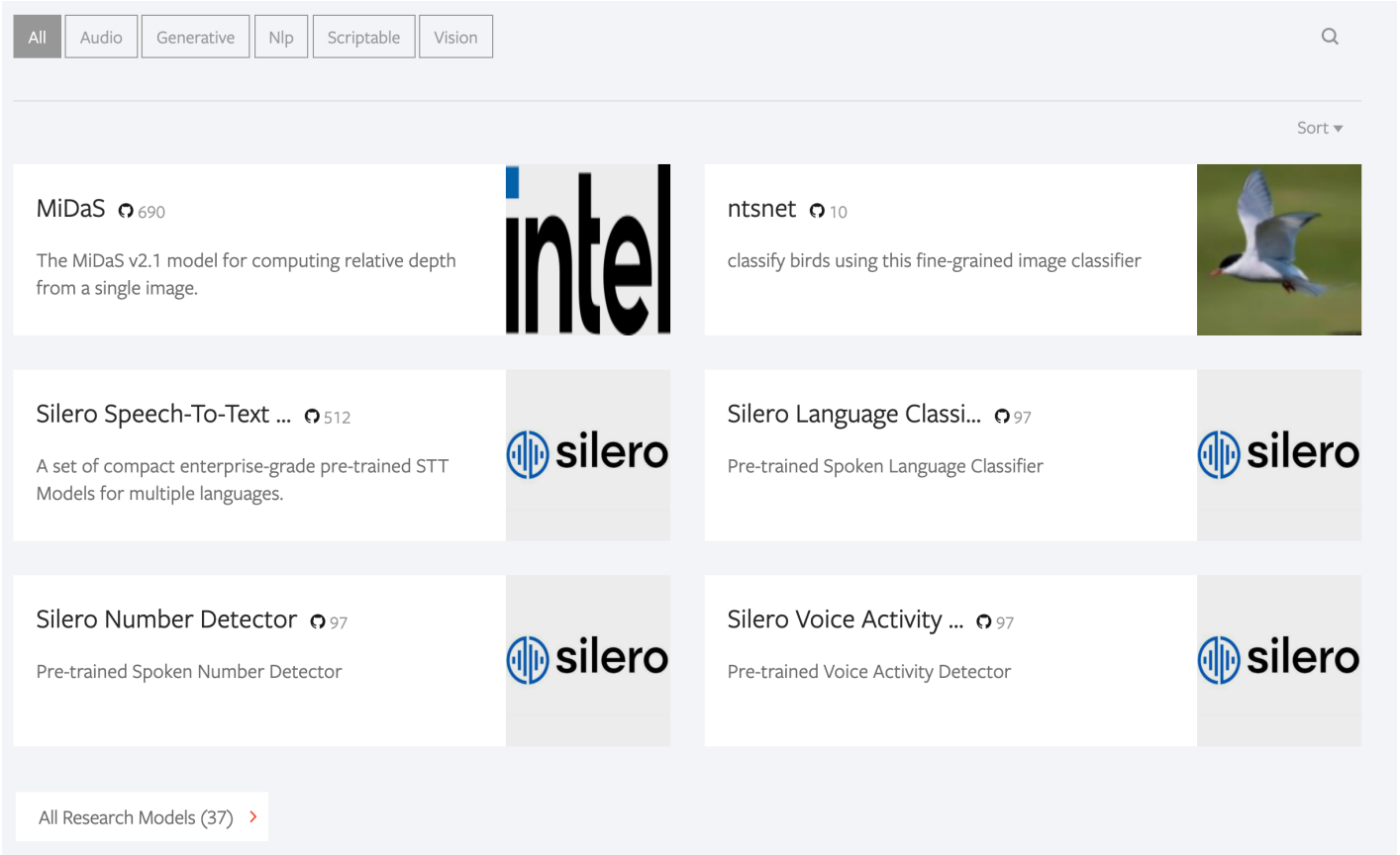
# 迁移学习范式


PyTorch
Get Started
Ecosystem
Mobile
Blog
Tutorials
Docs
Resources
Github
1.6.0
Search Tutorials
PyTorch Recipes
See All Recipes
Tutorials > Transfer Learning for Computer Vision Tutorial
Run in Google Colab
Download Notebook
View on GitHub
TRANSFER LEARNING FOR COMPUTER VISION TUTORIAL


IMAGENET
Pre-train
fc
conv3
conv2
conv1
Fine-tune
new fc
conv3
conv2
conv1

# 预训练模型选择

- 选择哪一个预训练模型更好？

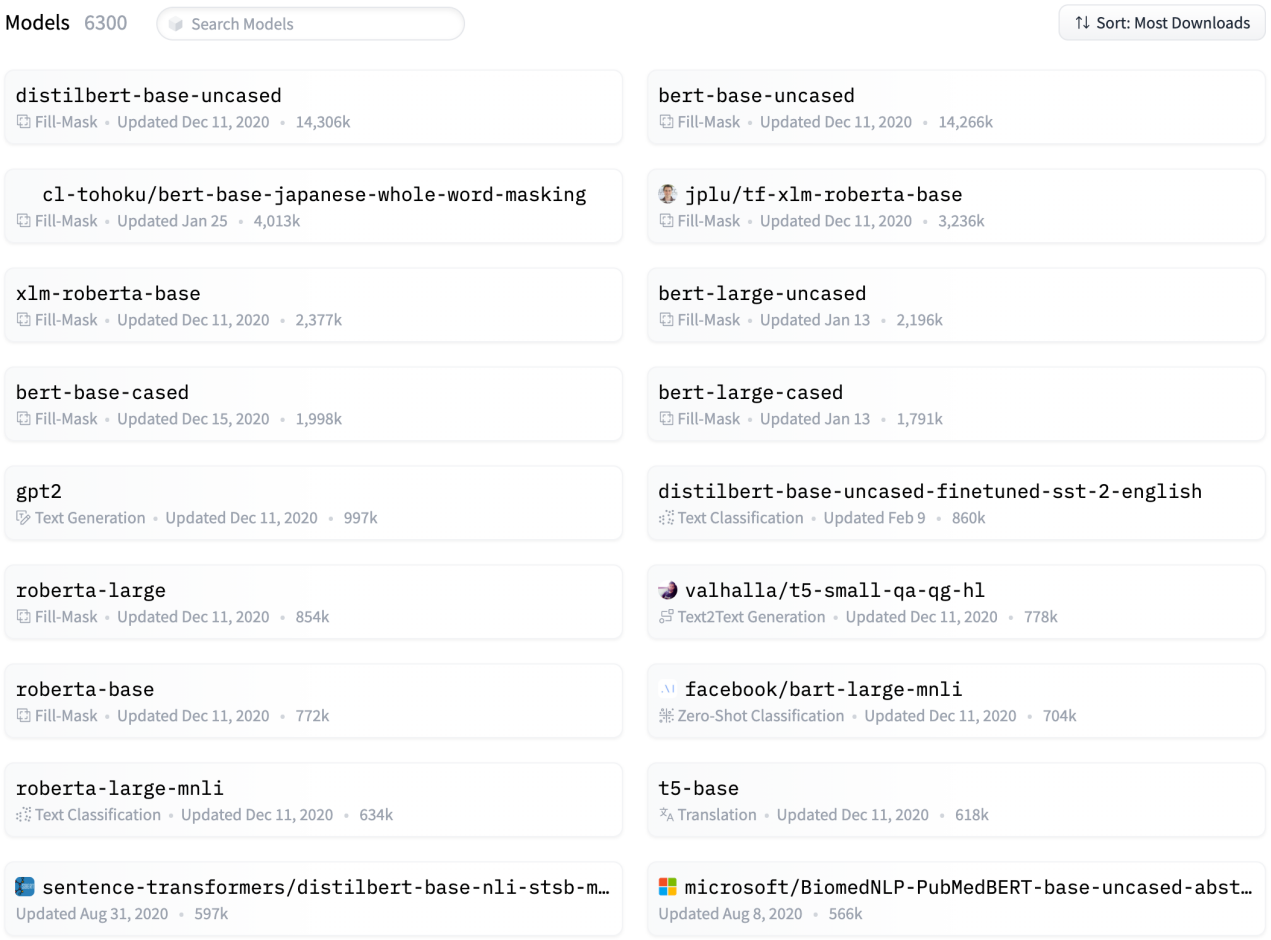



- AlexNet
- VGG
- ResNet
- SqueezeNet
- DenseNet
- Inception v3
- GoogLeNet
- ShuffleNet v2
- MobileNetV2
- MobileNetV3
- ResNeXt
- Wide ResNet
- MNASNet

- https://pytorch.org/hub/
- https://pytorch.org/vision/stable/models.html

# 预训练模型选择

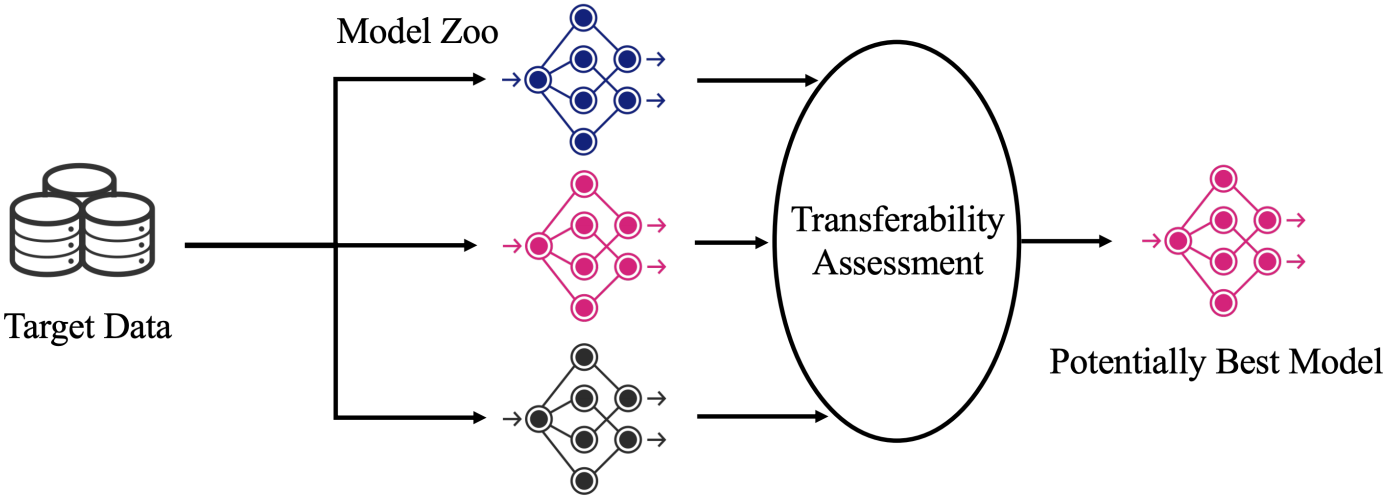
- 选择哪一个预训练模型更好？




- https://huggingface.co/models

# 预训练模型选择

- 预训练模型评估流程




- 寻找实用的评估指标
  - 通用
  - 快速
  - 准确

# 寻找实用的评估指标

- 通用
  - 应用场景广

| Modality | Pre-train | Target |
|---|---|---|
| vision | classification | classification |
| | classification | regression |
| | contrastive | classification |
| | contrastive | regression |
| language | LM | classification |

- 快速
  - 用时间来衡量
- 准确
  - 与实际指标的相关性

# 如何衡量评估指标的准确性

- Problem Setup
  - M pre-trained models $\{\phi_m\}_{m=1}^M$ with a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$
  - Ground truth transfer learning performance $\{T_m\}_{m=1}^M$
  - Assessment score $\{S_m\}_{m=1}^M$
- 线性相关系数？
  - 数值无直观含义
  - 无法处理非线性情况（s 与 log s）
- 评估指标的目的
  - 模型选择 $S_i > S_j \implies T_i > T_j$
- 好的评估指标
  - 基于逆序对个数

# Kendall Tau

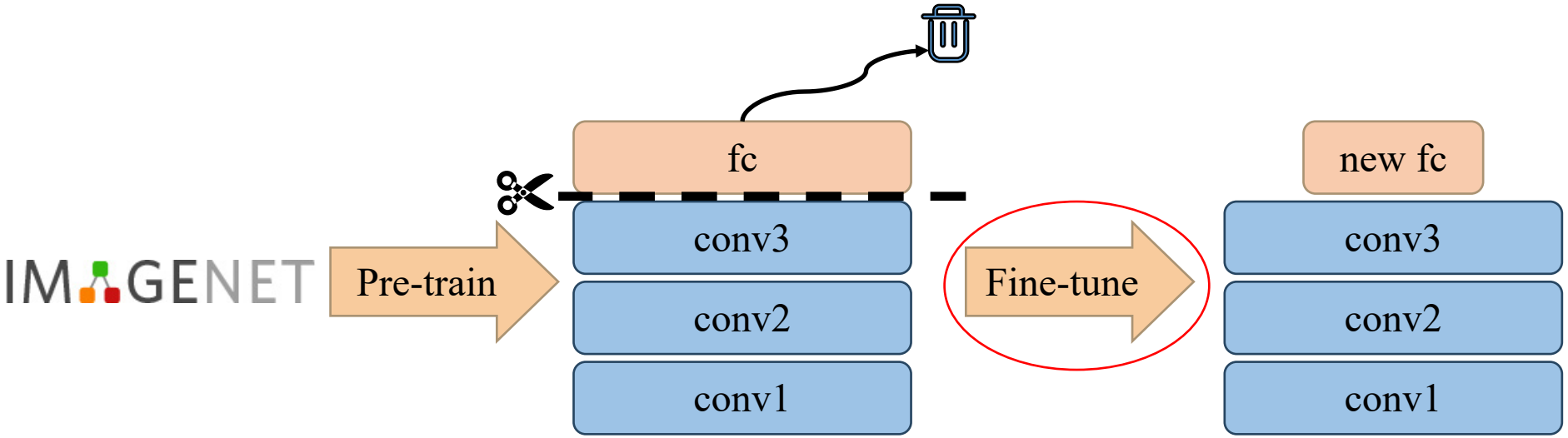
- Kendall Tau 系数

$$\tau = \frac{2}{M(M-1)} \sum_{1 \le i < j \le M} \mathrm{sgn}(T_i - T_j)\mathrm{sgn}(S_i - S_j)$$

  - sgn 函数取值1 / -1
  - 顺序对： $\mathrm{sgn}(T_i - T_j)\mathrm{sgn}(S_i - S_j) = 1$
  - 逆序对： $\mathrm{sgn}(T_i - T_j)\mathrm{sgn}(S_i - S_j) = -1$
  - $\tau = 1$ ： $S_i > S_j \iff T_i > T_j$ (perfect)
  - $\tau = -1$： $S_i > S_j \iff T_i < T_j$ (bad?)
  - 一般来说， $S_i > S_j \implies T_i > T_j$ (with probability $\frac{\tau + 1}{2}$)
  - 理想情况： consistently $\tau = 1$
- 加权Kendall Tau 系数
  - 关注S或者T大的(T,S)对

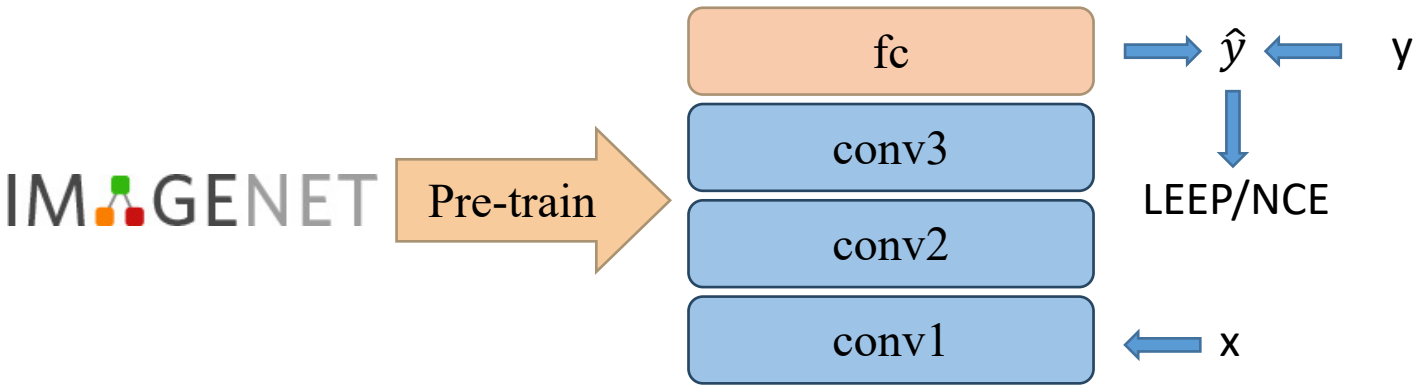
$$S_i > S_j \wedge S_j \text{ is large} \implies T_i > T_j \text{ (with probability } > \frac{\tau_w + 1}{2})$$

# 暴力搜索



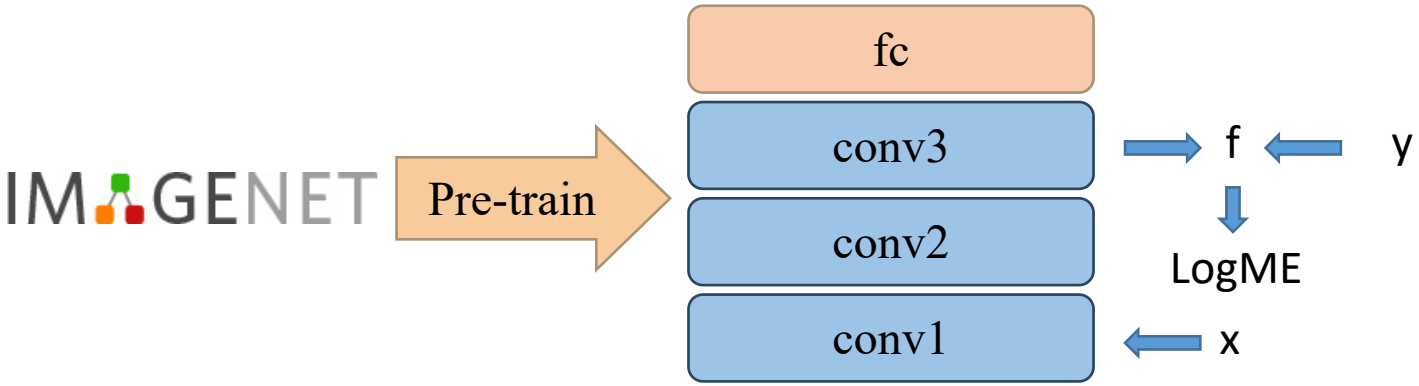

- 完成整个finetune流程
  - 超参数调优、模型训练（☹ 时间长）
  - 得到ground-truth评估指标（☺ $\tau_w = 1$，选择准）
  - 应用范围广（☺ 通用）
- 具体时间
  - 对两个超参数（学习率与L2衰减系数）各自进行7次grid search
  - 为了评估一个模型的迁移效果，需要 49 * 1hr ≈ 50 hr

# LEEP/NCE




- 不更新预训练模型，建模类别关系
  - 没有训练过程（☺ 时间极短）
  - 选择效果较差（☹ 实验测得$\tau_w$较小）
  - 应用范围受限
    - ☹ 只适用于有监督预训练模型迁移到分类任务的场景
- 评估过程中引入不必要的预训练模型的fc
  - LEEP/NCE与预训练fc相关，迁移性指标与预训练fc无关

# LogME设计思路




- 不更新预训练模型⇒时间快☺
- 只使用预训练模型的特征提取器
  - 可适用于任何方式训练的预训练模型☺
- 问题转化
  - 如何衡量预训练特征f与标注y的关系

# LogME设计思路

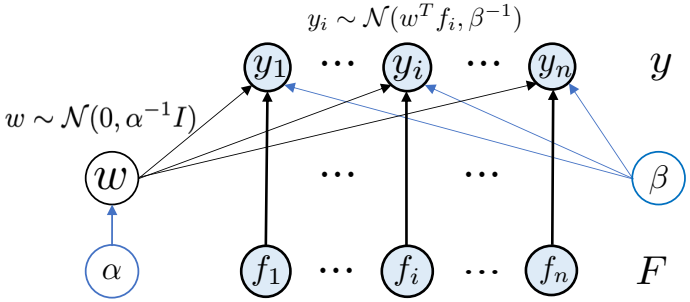
- 如何衡量预训练特征f与标注y的关系
  - 概率密度p(y|f)
  - 关系建模 $y=w^{T}f$
- 点估计
  - 训练得到最优w*，计算p(y|f, w*)
  - 容易过拟合（本身就是一个监督学习问题，有超参数）
- 分布估计
  - 利用w的分布p(w)，遍历w的可能取值

$$p(y|F)=\int p(w)p(y|F,w)\mathrm{d}w$$

  - 文献中称为evidence
  - posterior $=\frac{likelihood*prior}{evidence}$

# LogME具体推导——一元回归

- 概率图模型

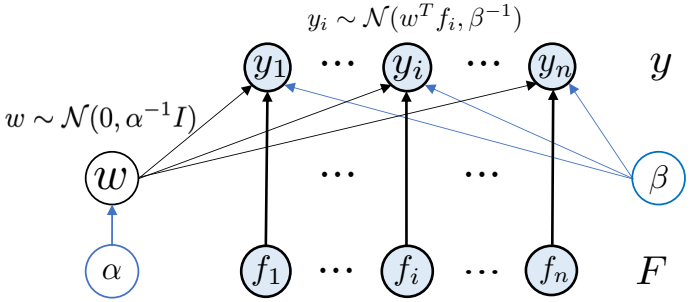



- evidence $p(y|f,\alpha,\beta)$

$$
\begin{aligned}
p(y|F,\alpha,\beta) &= \int p(w|\alpha)p(y|F,w,\beta)\mathrm{d}w \\
&= \int p(w|\alpha)\prod_{i=1}^{n} p(y_i|f_i,w,\beta)\mathrm{d}w \\
&= (\frac{\beta}{2\pi})^{\frac{n}{2}}(\frac{\alpha}{2\pi})^{\frac{D}{2}} \int e^{-\frac{\alpha}{2}w^T w-\frac{\beta}{2}||Fw-y||^2}\mathrm{d}w
\end{aligned}
$$

# LogME具体推导——一元回归

- 概率图模型



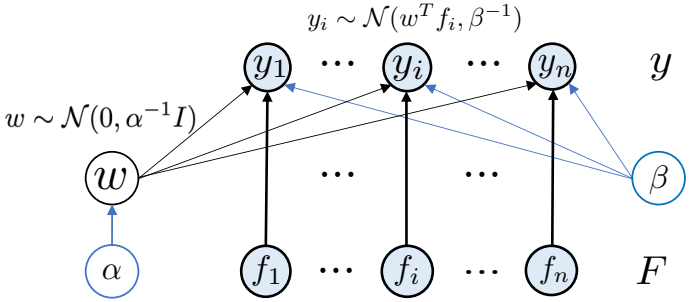

- evidence $p(y|f, \alpha, \beta)$

$$\begin{aligned}\mathcal{L}(\alpha,\beta) &= \log p(y|F,\alpha,\beta) \\ &= \frac{n}{2}\log\beta + \frac{D}{2}\log\alpha - \frac{n}{2}\log 2\pi \\ &- \frac{\beta}{2}||Fm-y||_2^2 - \frac{\alpha}{2}m^T m - \frac{1}{2}\log|A|\end{aligned}$$

$$A = \alpha I + \beta F^T F, m = \beta A^{-1} F^T y.$$

# LogME具体推导——一元回归

- 概率图模型




- 如何选择$\alpha, \beta$的值？
  - 无需grid search!
  - 交替优化

$$A = \alpha I + \beta F^T F, m = \beta A^{-1} F^T y, \gamma = \sum_{i=1}^{D} \frac{\beta \sigma_i}{\alpha + \beta \sigma_i}$$

$$\alpha \leftarrow \frac{\gamma}{m^T m}, \beta \leftarrow \frac{n - \gamma}{||Fm - y||_2^2}$$

  - 求得最大化对数证据(LogME, log maximum evidence)

$$(\alpha^*, \beta^*) = \arg\max_{\alpha, \beta} \mathcal{L}(\alpha, \beta)$$

# LogME具体推导——更多场景

- 多元回归
  - 每一个维度分别计算LogME，再求平均
- 分类问题（K类）
  - 采用softmax输出，则evidence无解析表达式
  - 转化为K个二分类问题，二分类问题视作0-1回归
  - 等价于回归one-hot labels

# LogME具体流程

**Algorithm 1** LogME

**Input:** Pre-trained model $\phi$
Target dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$
**Output:** logarithm of maximum evidence (LogME)
Extract features using pre-trained model $\phi$:
$F \in \mathbb{R}^{n \times D}, f_i = \phi(x_i), Y \in \mathbb{R}^{n \times K}$
Compute SVD $F^T F = V \text{diag}\{\sigma\} V^T$
**for** $k = 1$ to $K$ **do**
  Let $y = Y^{(k)} \in \mathbb{R}^n$, initialize $\alpha = 1, \beta = 1$
  **while** $\alpha, \beta$ not converge **do**
    Compute $\gamma = \sum_{i=1}^{D} \frac{\beta\sigma_i}{\alpha + \beta\sigma_i}, \Lambda = \text{diag}\{(\alpha + \beta\sigma)\}$
    **Naïve**: $A = \alpha I + \beta F^T F, m = \beta A^{-1} F^T y$
    **Optimized**: $m = \beta(V(\Lambda^{-1}(V^T(F^T y))))$
    Update $\alpha \leftarrow \frac{\gamma}{m^T m}, \beta \leftarrow \frac{n - \gamma}{||Fm - y||_2^2}$
  **end while**
  Compute $\mathcal{L}_k = \frac{1}{n}\mathcal{L}(\alpha, \beta)$ using Eq. 2
**end for**
Return LogME $\frac{1}{K}\sum_{k=1}^{K} \mathcal{L}_k$

计算复杂度 $\mathcal{O}(KD^3 + nKD^2)$

对于常见情况

$D \approx 10^3, n \approx 10^4, K \approx 10^3$

计算次数约$10^{12}$
CPU频率GHz
计算时间约$10^3$秒

不够快！

计算瓶颈：
第9行矩阵乘矩阵，矩阵求逆

# LogME算法优化

**Algorithm 1** LogME

1: **Input:** Pre-trained model $\phi$
Target dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$
2: **Output:** logarithm of maximum evidence (LogME)
3: Extract features using pre-trained model $\phi$:
$F \in \mathbb{R}^{n \times D}, f_i = \phi(x_i), Y \in \mathbb{R}^{n \times K}$
4: Compute SVD $F^T F = V \text{diag}\{\sigma\} V^T$
5: **for** $k = 1$ to $K$ **do**
6: Let $y = Y^{(k)} \in \mathbb{R}^n$, initialize $\alpha = 1, \beta = 1$
7: **while** $\alpha, \beta$ not converge **do**
8: Compute $\gamma = \sum_{i=1}^{D} \frac{\beta\sigma_i}{\alpha + \beta\sigma_i}, \Lambda = \text{diag}\{(\alpha + \beta\sigma)\}$
9: **Naïve**: $A = \alpha I + \beta F^T F, m = \beta A^{-1} F^T y$
10: **Optimized**: $m = \beta(V(\Lambda^{-1}(V^T(F^T y))))$
11: Update $\alpha \leftarrow \frac{\gamma}{m^T m}, \beta \leftarrow \frac{n - \gamma}{||Fm - y||_2^2}$
12: **end while**
13: Compute $\mathcal{L}_k = \frac{1}{n}\mathcal{L}(\alpha, \beta)$ using Eq. 2
14: **end for**
15: Return LogME $\frac{1}{K}\sum_{k=1}^{K} \mathcal{L}_k$

计算瓶颈：

第9行矩阵乘矩阵，矩阵求逆

$$D \approx 10^3, n \approx 10^4, K \approx 10^3$$

加速方法：充分利用第4行分解结果

$$\Lambda = \text{diag}\{(\alpha + \beta\sigma)\}$$
$$A = \alpha I + \beta F^T F = V \Lambda V^T$$
$$A^{-1} = V \Lambda^{-1} V^T$$

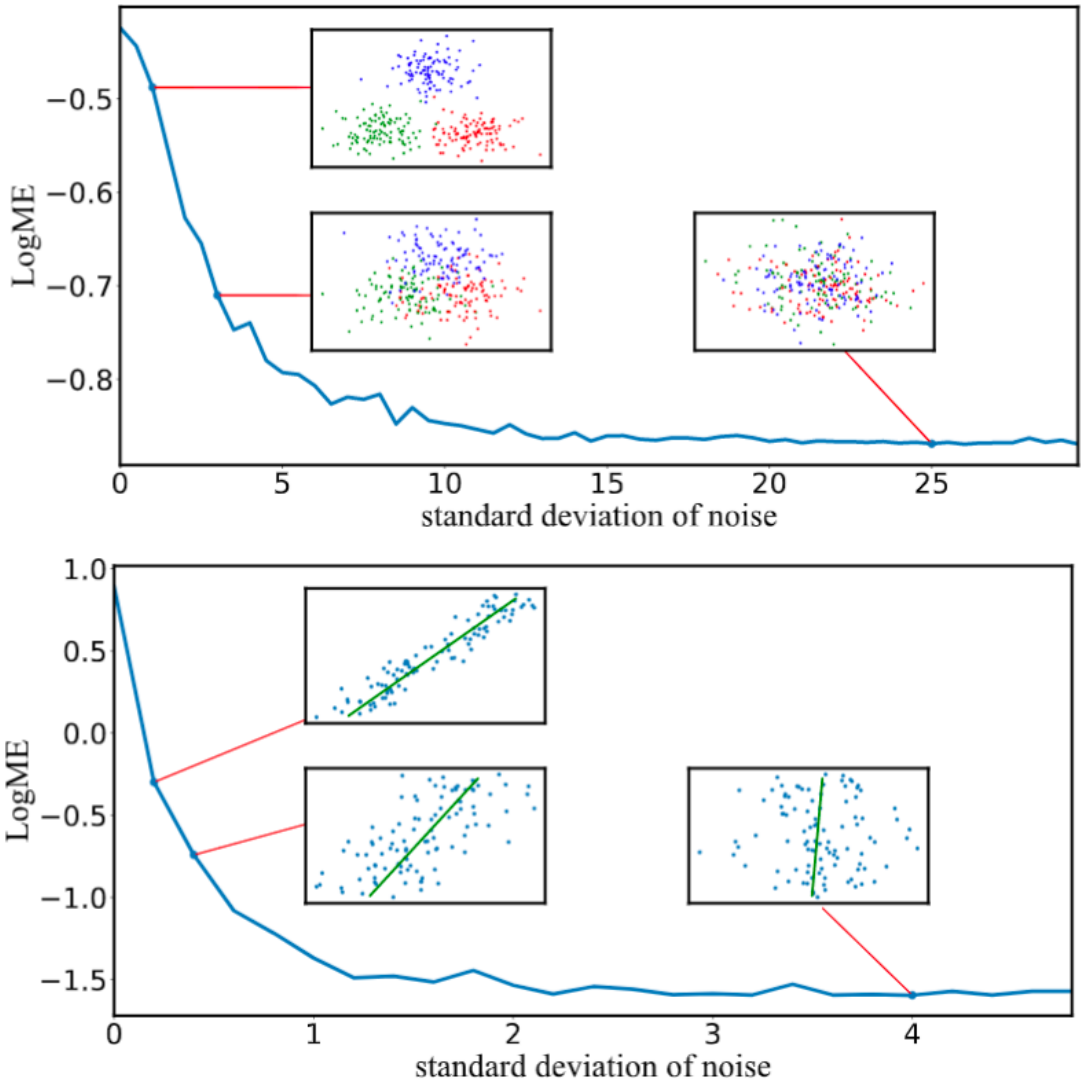
$$A^{-1} F^T y = (V(\Lambda^{-1}(V^T(F^T y))))$$

不需要矩阵求逆
矩阵-矩阵乘法 → 矩阵-向量乘法

总体复杂度从四次方降低为三次方

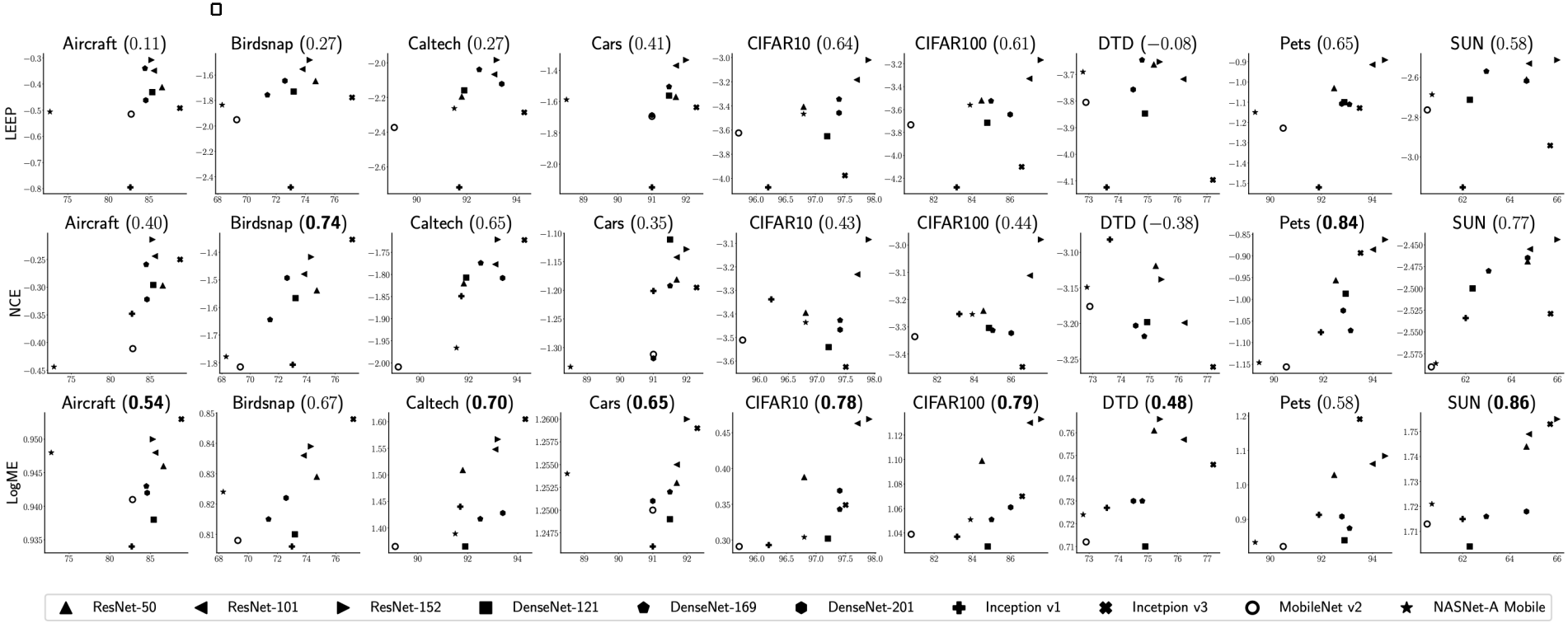
| | Complexity per for-loop | Overall complexity |
|---|---|---|
| naïve | $\mathcal{O}(D^3 + nD^2)$ | $\mathcal{O}(KD^3 + nKD^2)$ |
| optimized | $\mathcal{O}(D^2 + nD)$ | $\mathcal{O}(KD^2 + nKD + D^3 + nD^2)$ |

# 实验——人造数据



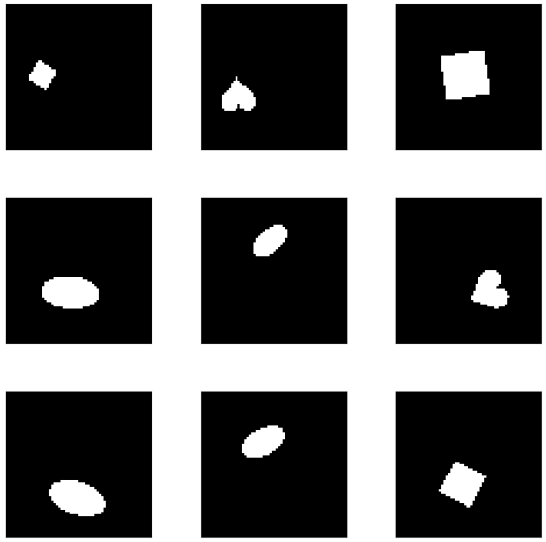

直观感受LogME随着特征质量的变化情况

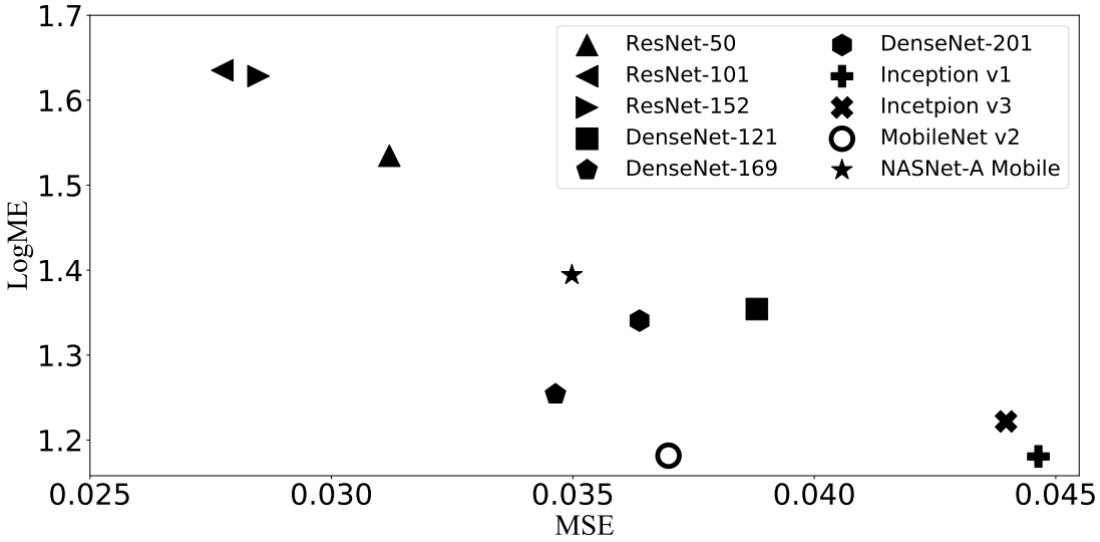
# 实验——有监督预训练、分类任务




- 9个数据集，10个预训练模型
- 横轴迁移准确率，纵轴评估指标
- LogME的$\tau_w$在大部分任务上是最高的
- 大部分情况下$\tau_w \approx 0.7/0.8$，选择准确率85%/90%

# 实验——有监督预训练、回归任务





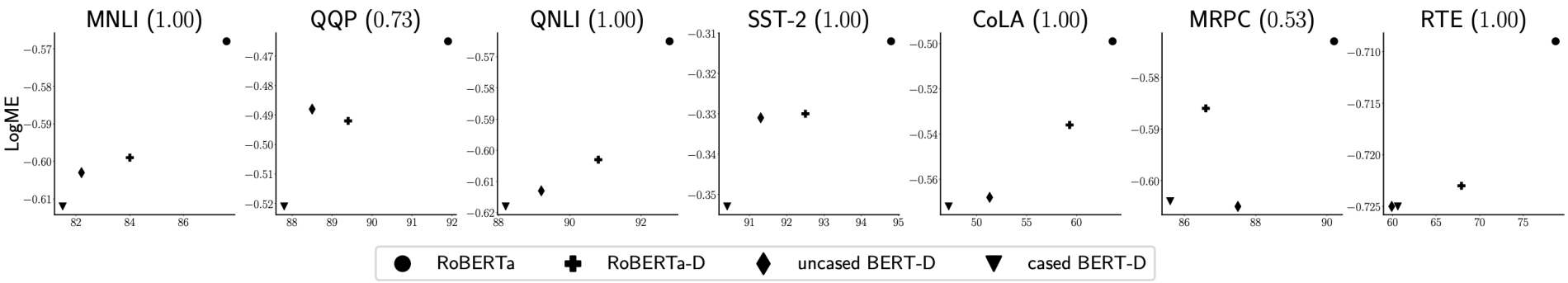

- dSprites数据集，10个预训练模型
- 横轴迁移指标（MSE↓），纵轴LogME↑
  - 只有LogME能处理回归
- MSE与LogME有明显负相关，结果符合预期

# 实验——无监督预训练模型

| Pre-trained Network | Aircraft | | dSprites | |
|---|---|---|---|---|
| | Accuracy (%) | LogME | MSE | LogME |
| MoCo V1 | 81.68 | 0.93 | 0.069 | 1.52 |
| MoCo V2 | 84.16 | 0.94 | 0.047 | 1.64 |
| MoCo 800 | 86.99 | 0.95 | 0.050 | 1.58 |
| | $\tau_w$: 1.0 | | $\tau_w$: 1.0 | |

- 分类回归任务均能完美预测预训练模型的好坏
- 预训练模型的排序是任务相关的

# 实验——自然语言处理




- GLUE benchmark中的七个任务
- 四个最受欢迎（下载量最大）的预训练模型
- HuggingFace Model Hub上面report的准确率
- LogME完美预测了五个任务上的预训练模型排序

# 实验——耗时分析

| | Wall-clock time (second) | Proportion |
|---|---|---|
| fine-tune (upper bound) | $(1.61 \pm 0.06) \times 10^5$ | 1000‰ |
| extract feature (lower bound) | $37.3 \pm 0.6$ | 0.23‰ |
| LEEP (Nguyen et al., 2020) | $37.3 \pm 0.6$ | 0.23‰ |
| NCE (Tran et al., 2019) | $37.5 \pm 0.6$ | 0.23‰ |
| LogME (naïve implementation) | $839.8 \pm 5.6$ | 5.22‰ |
| LogME (optimized) | $50.4 \pm 0.7$ | 0.31‰ |

- LEEP/NCE耗时接近下界，但是不准确、适用范围窄
- LogME的简单实现，效果好，但是耗时较长
- 优化之后的LogME，耗时接近下界
  - 不到一分钟
  - 加速3000倍

# LogME使用

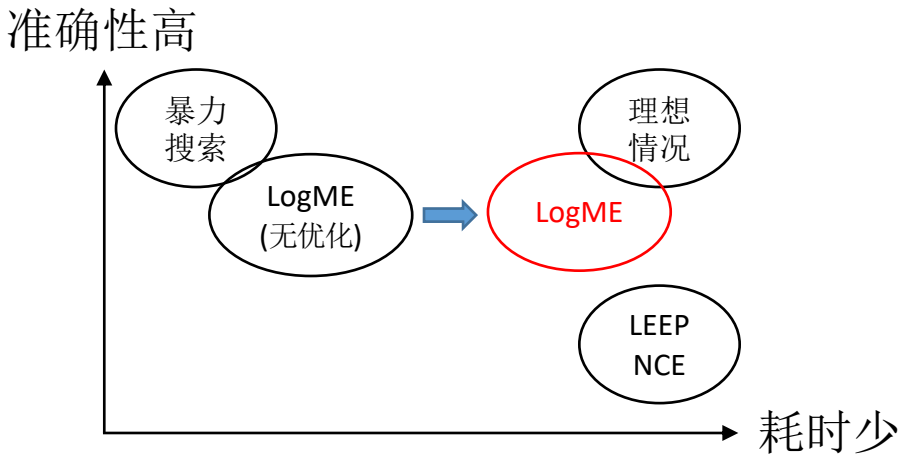
- 安装依赖项
  - torch, numpy, numba
- 准备好特征f和标注y
  - 注明是否是回归问题。分类时y是整数，回归时y是小数
- 调用函数即可得到score
  - 根据score大小选择最好的预训练模型（LogME最大的模型）

```
def LogME(f: torch.Tensor, y: torch.Tensor, regression=False):
    """
    :param f: [N, F], feature matrix from pre-trained model
    :param y: target labels.
        For classification, y has shape [N] with element in [0, C_t).
        For regression, y has shape [N, C] with C regression-labels
    :param regression: whether regression
    :return: LogME score (how well f can fit y directly)
    """
```

https://github.com/thuml/LogME/blob/main/LogME.py

- 调用者无需计算$\tau_w$

# 总结与展望




- 有理论保障的方法一般来说计算速度慢
- LogME的加速算法使得它既有理论保障又速度快
- 可能的应用场景
  - 预训练模型选择
  - 预训练过程中early stopping（e.g.自监督学习）
- 可能的改进方向
  - 检测、分隔等复杂任务的模型选择
  - 稍微牺牲时间效率，换取更高准确性

# Q&A

游凯超

2021.3.10