Deep Learning

Deeper, Better, _____ , Stronger than Machine Learning

Valentin Barriere

Universidad de Chile – DCC

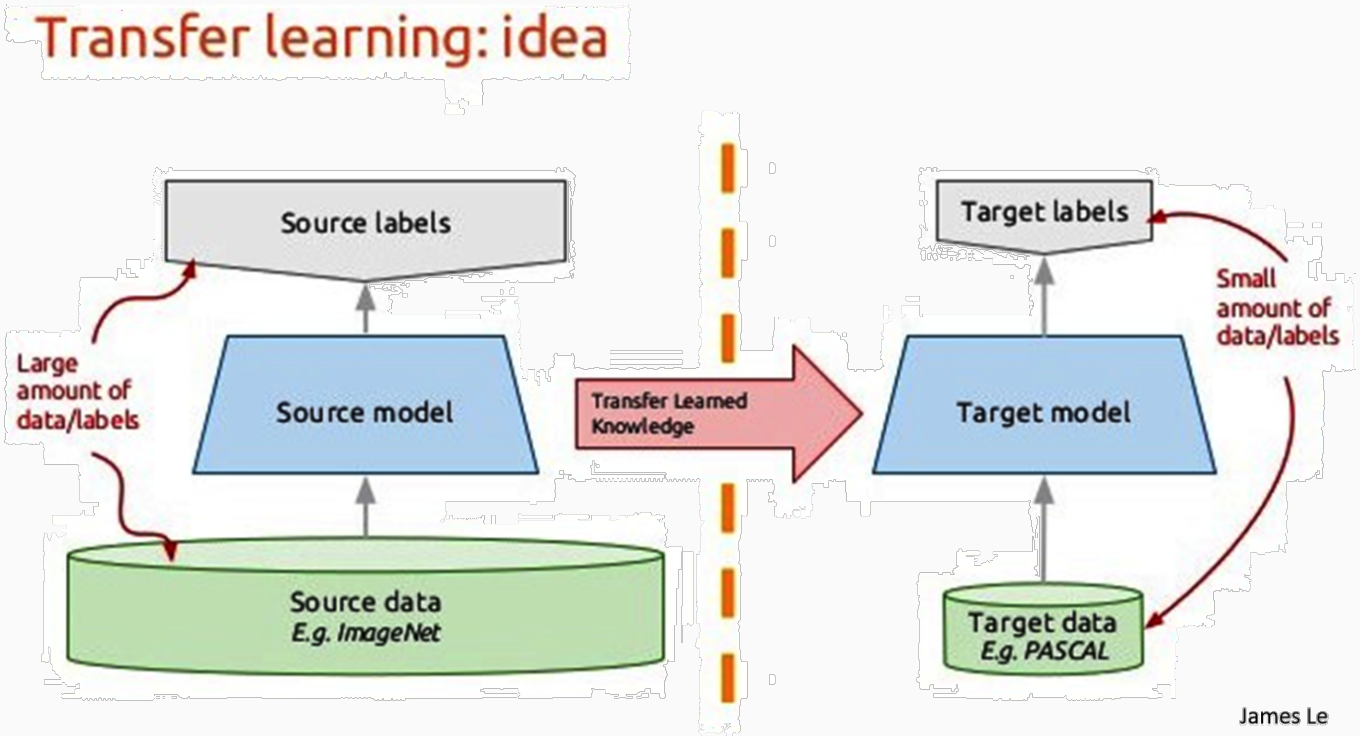
CC6204, Primavera 2024

Transfer Learning

Transfer Learning and Fine-tuning

Transfer Learning

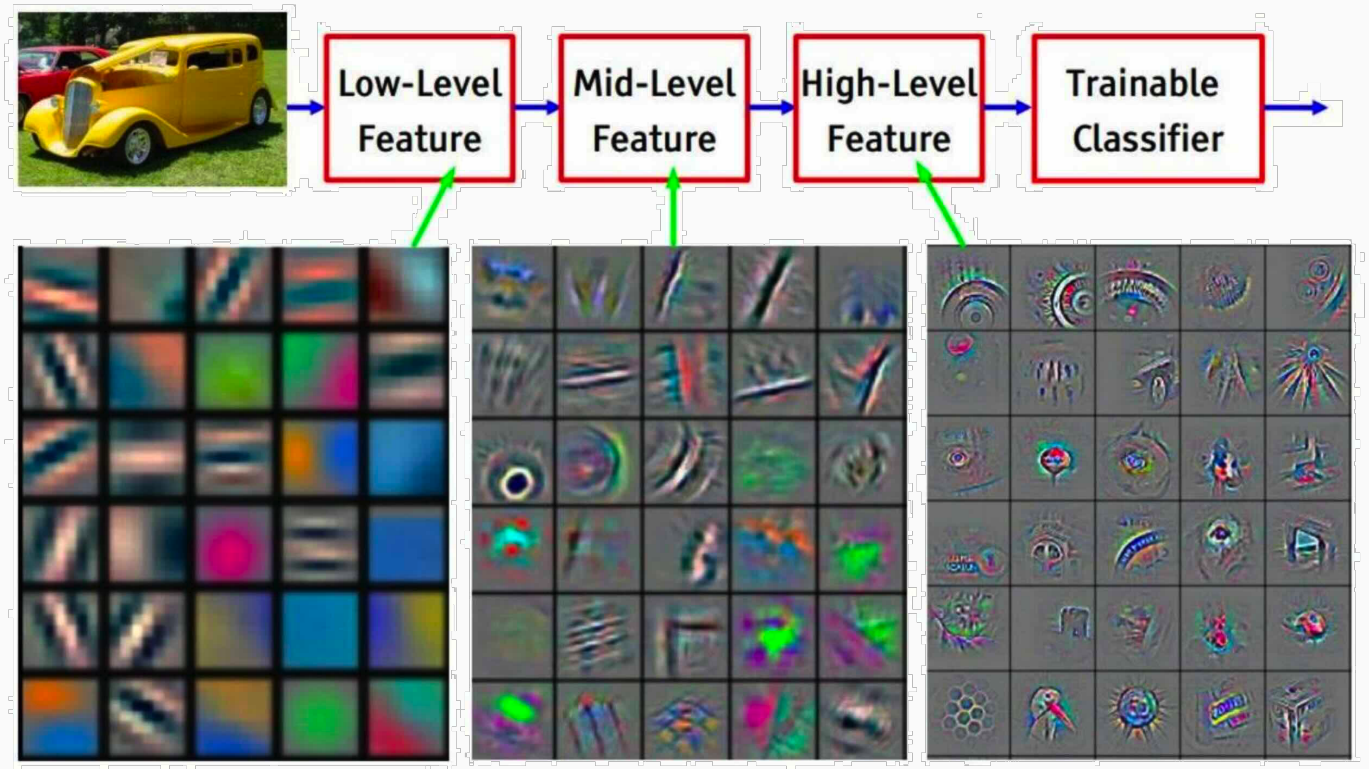
- Model trained on a large external dataset $\mathcal{D}_{ext}$ fine-tuned on a smaller dataset $\mathcal{D}$.
- Improves performance compared to training a model from scratch on $\mathcal{D}$.



Can be used in any domain: NLP, Computer Vision, Machine Listening, etc...

Motivations of TL

- **Limited Data**: Often, the target dataset is too small to train a deep model from scratch.
- **Faster Training**: Pre-trained models reduce the time and computational power required.
- **Higher Performance**: Models generalize better when they have learned useful features from large datasets.



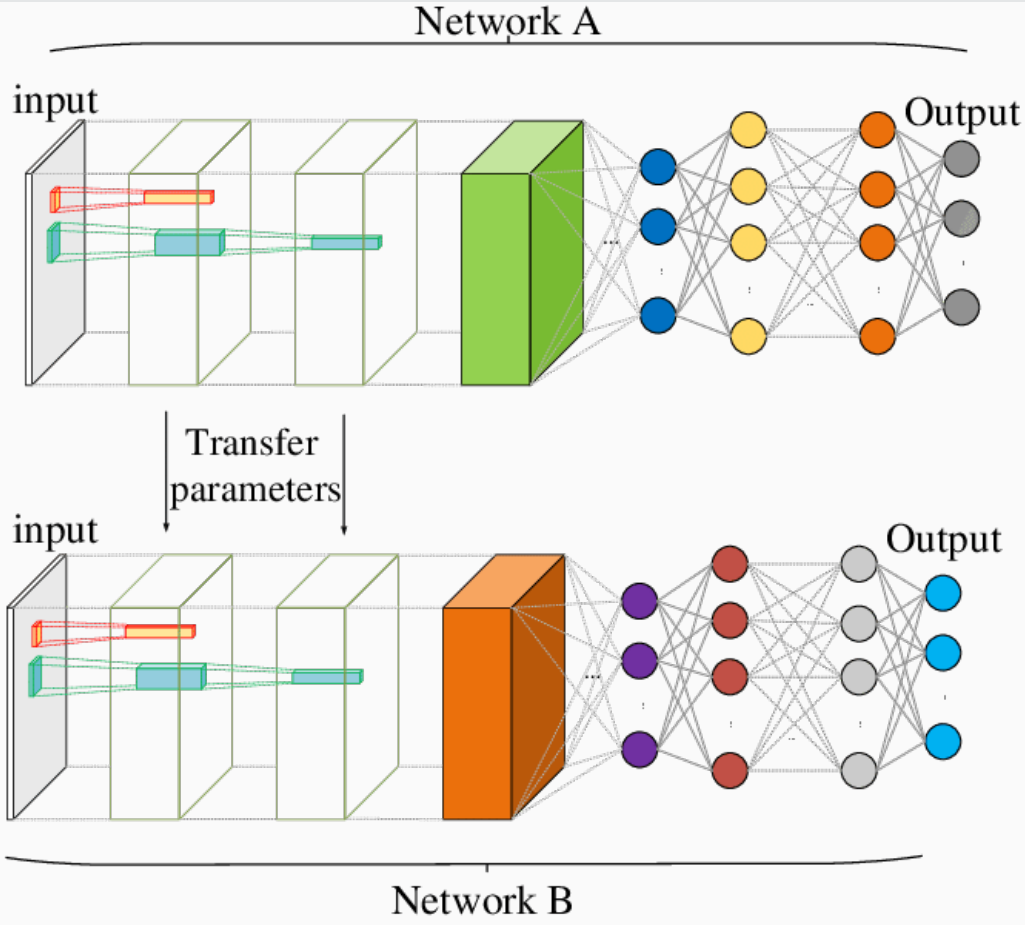
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Transfer Learning and Fine-tuning

Fine-tuning

Process of adjusting the model to the new dataset $\mathcal{D}$. It can involve:

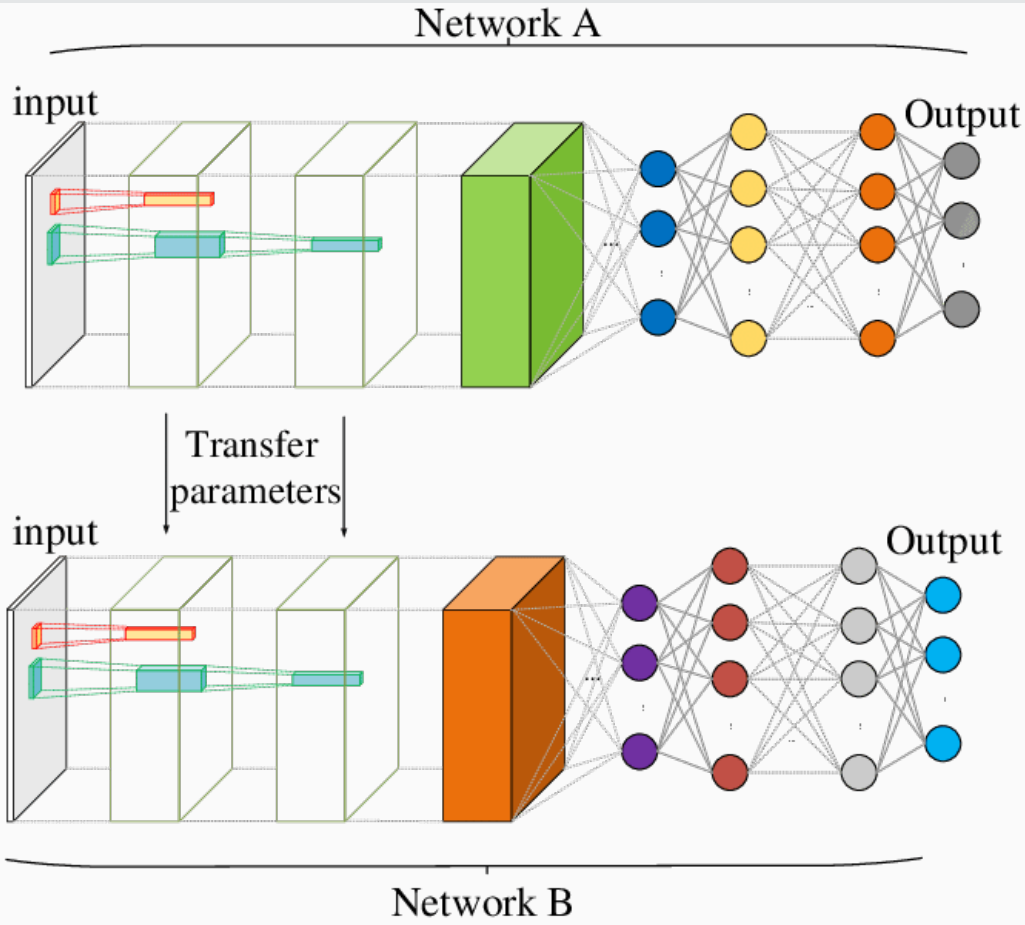
- re-training all layers
- only the deeper layers related to specific tasks (freezing the initial layers)
- re-training the deep layers and gradually unfreezing the initial layers



Types of Transfer Learning

Feature Extraction

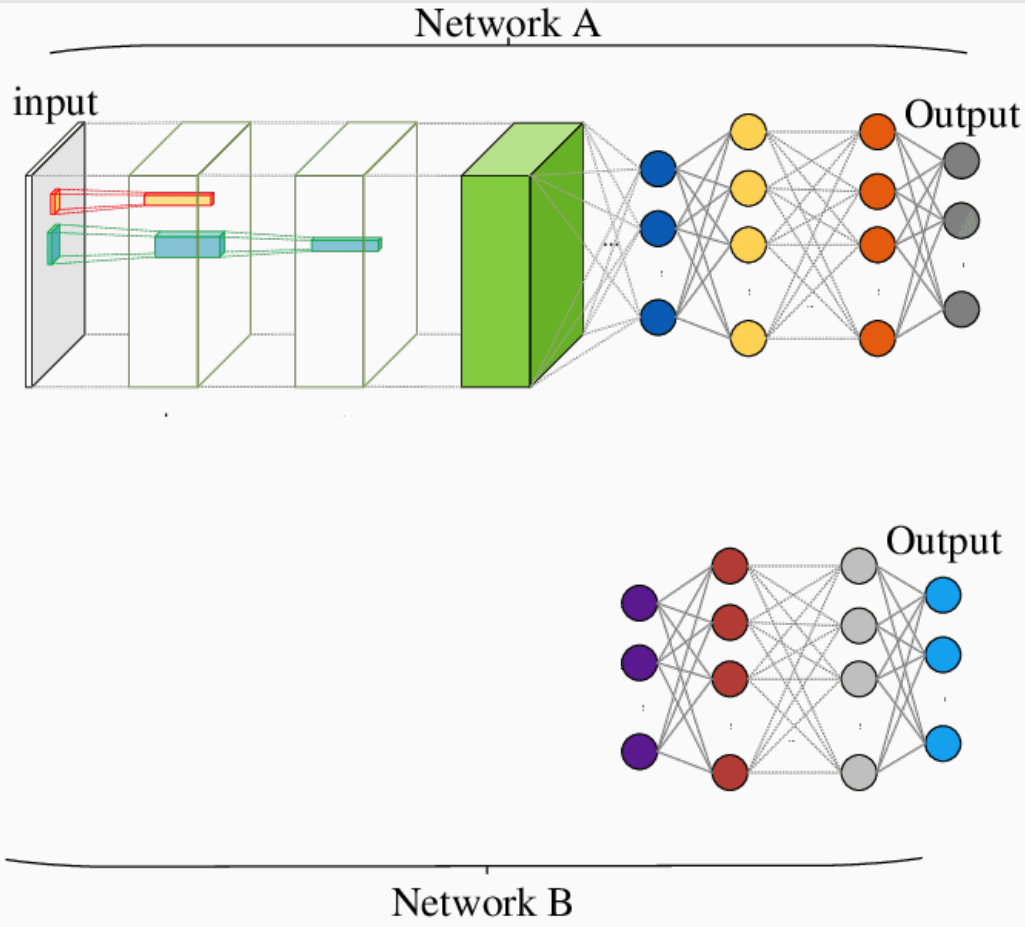
- **Method**: Freeze all layers of the pre-trained model, using it as feature extractor. Add new classifier layers on top and train only these layers.
- **Usage**: When the new dataset is small and does not require deep changes in the learned representations.



Types of Transfer Learning

Feature Extraction

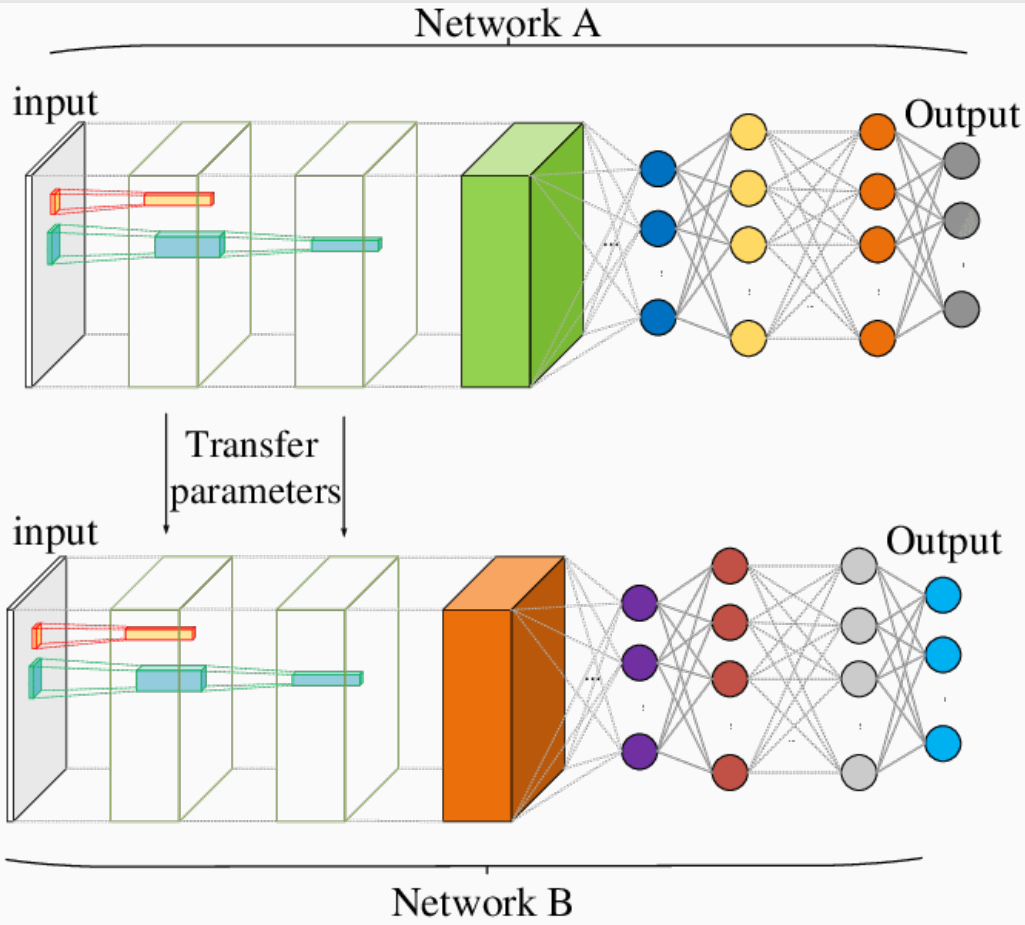
- **Method**: Freeze all layers of the pre-trained model, using it as feature extractor. Add new classifier layers on top and train only these layers.
- **Usage**: When the new dataset is small and does not require deep changes in the learned representations.



Types of Transfer Learning

Full Fine-tuning

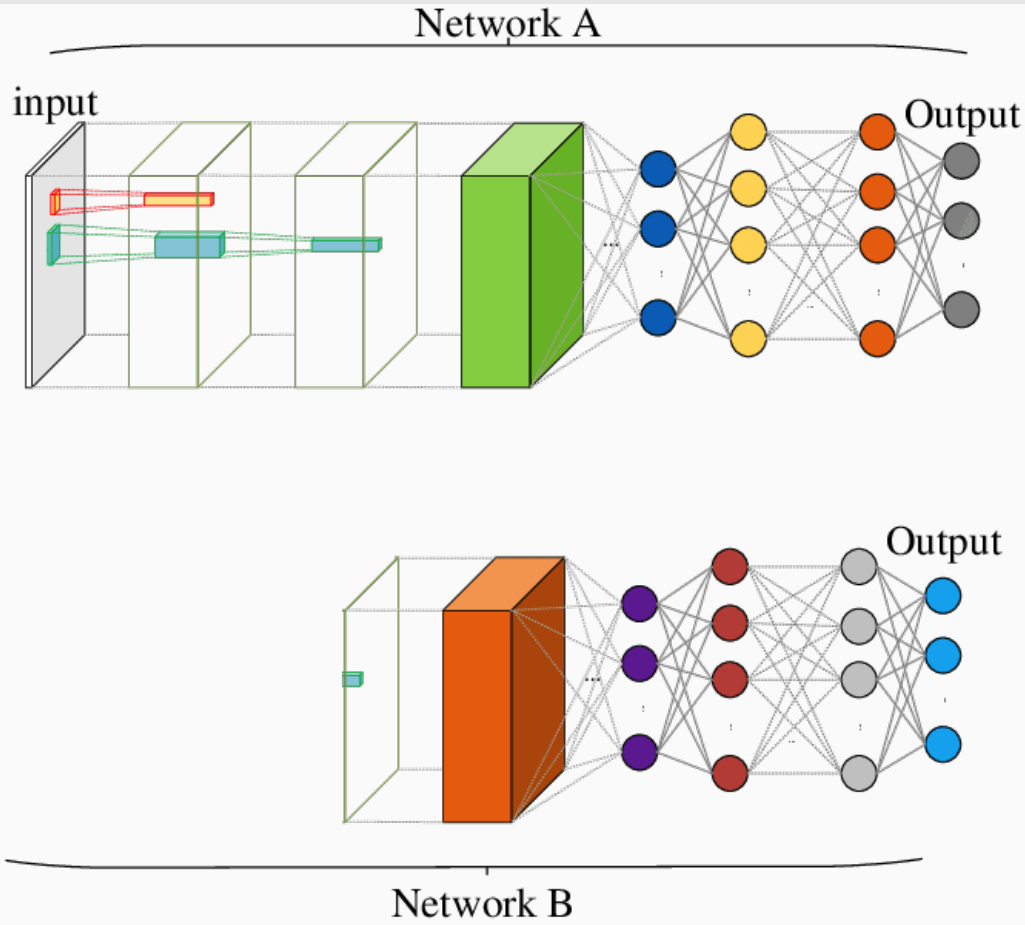
- **Method**: Fine-tune the whole pre-trained model by unfreezing all layers and retraining them on new task and data.
- **Usage**: When the new dataset is larger and more complex or when the tasks are very different.



Types of Transfer Learning

In Between: Trade-off Overfitting vs New features learning

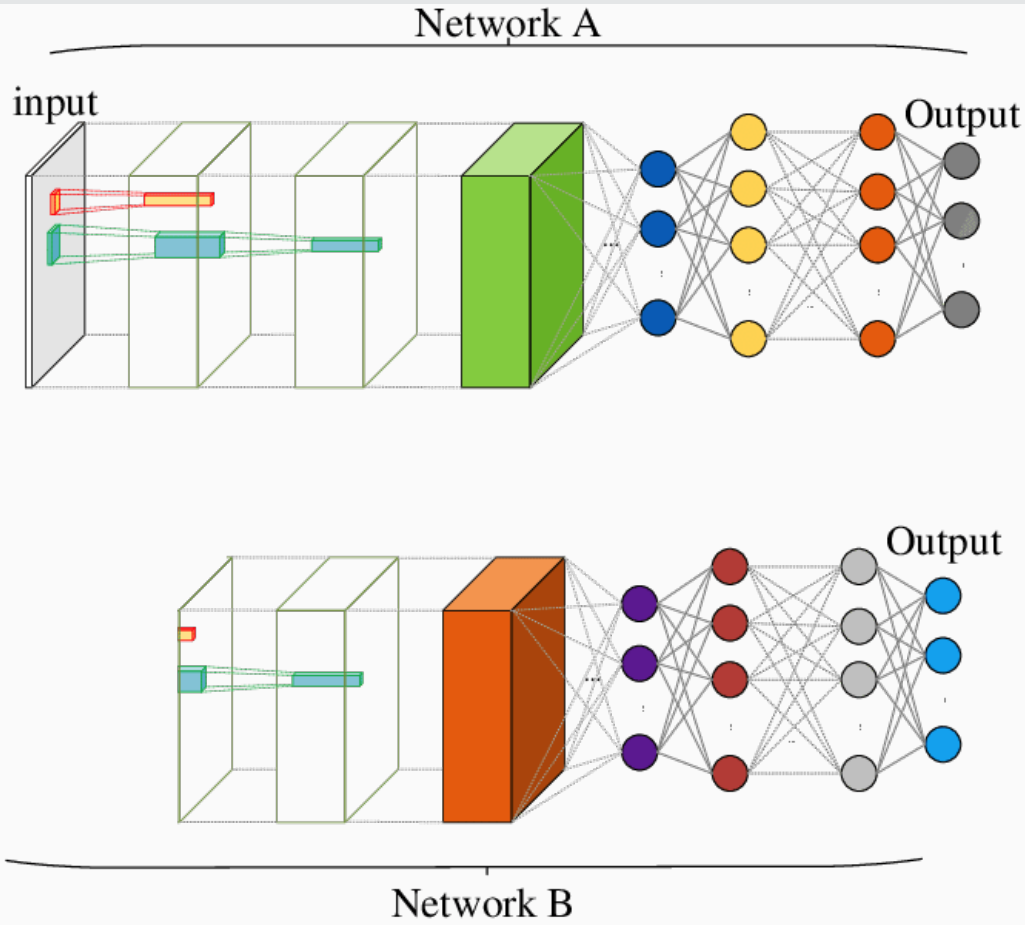
- **Method**: Freeze early layers (capture general basic features), fine-tune the deeper layers (task-specific features). Trade-off regarding depth.
- **Usage**: For datasets where the new task has similar low-level patterns but requires different higher-level representations.



Types of Transfer Learning

In Between: Trade-off Overfitting vs New features learning

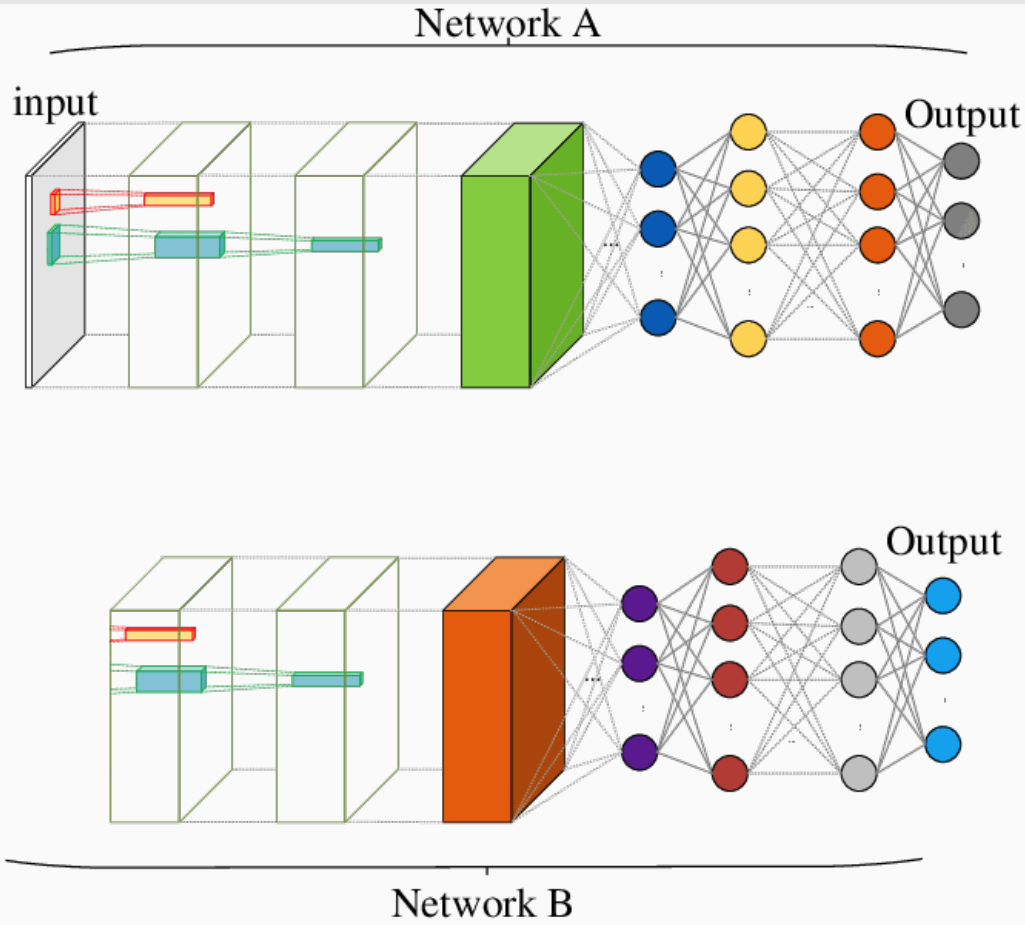
- **Method**: Freeze early layers (capture general basic features), fine-tune the deeper layers (task-specific features). Trade-off regarding depth.
- **Usage**: For datasets where the new task has similar low-level patterns but requires different higher-level representations.



Types of Transfer Learning

In Between: Trade-off Overfitting vs New features learning

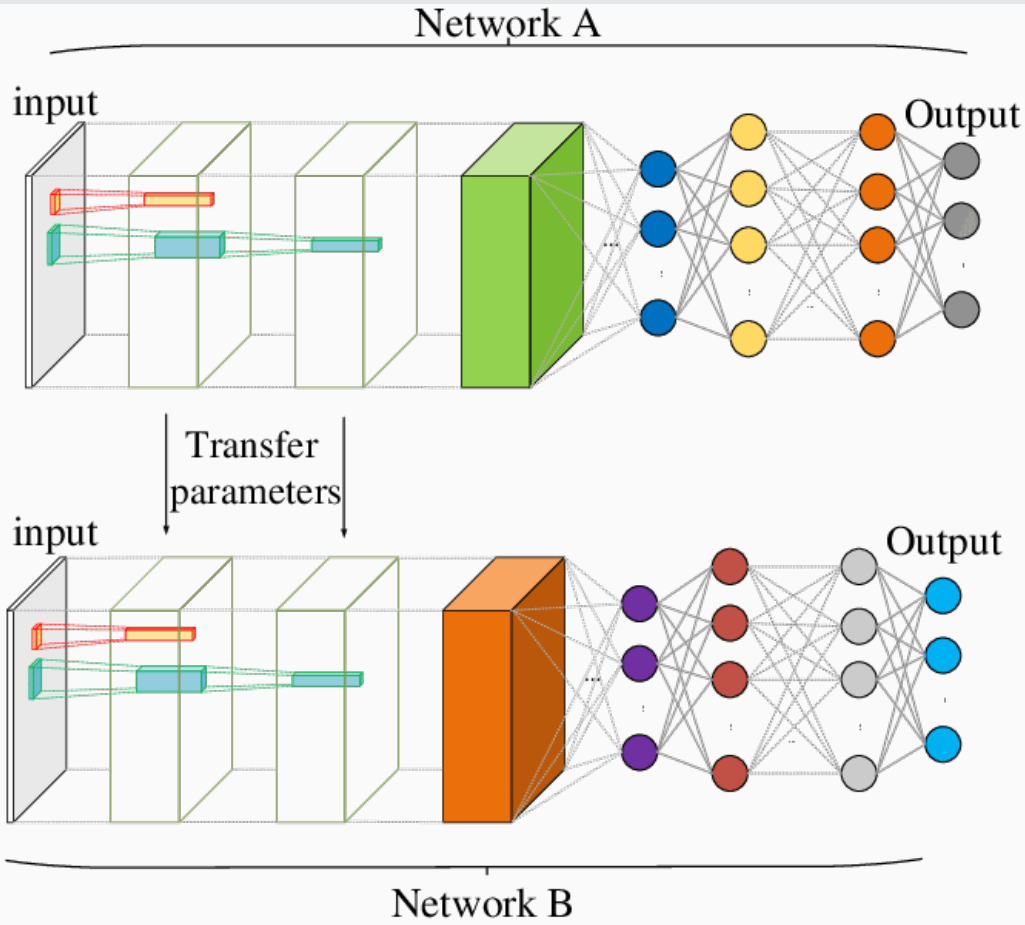
- **Method**: Freeze early layers (capture general basic features), fine-tune the deeper layers (task-specific features). Trade-off regarding depth.
- **Usage**: For datasets where the new task has similar low-level patterns but requires different higher-level representations.



Types of Transfer Learning

In Between: Trade-off Overfitting vs New features learning

- **Method**: Freeze early layers (capture general basic features), fine-tune the deeper layers (task-specific features). Trade-off regarding depth.
- **Usage**: For datasets where the new task has similar low-level patterns but requires different higher-level representations.



Simple Transfer Learning

Tutorial on Transfer Learning using Keras:

Figure 1: Examples of special dataset/task to fine-tuned a model on

A network pre-trained over ImageNet would be perfect for this type of task and data

Pre-trained Models

Advantages: These models capture rich feature representations and can be reused for similar tasks.

- **Image Models**: VGG, ResNet, Inception, EfficientNet (pre-trained on ImageNet), ViT, SimCLR (pre-trained on massive unlabeled images dataset)
- **NLP Models**: BERT, GPT, RoBERTa (pre-trained on massive text corpora).
- **Audio Models**: Wav2Vec 2.0, VGGish, DeepSpeech (pre-trained on LibriSpeech, AudioSet).

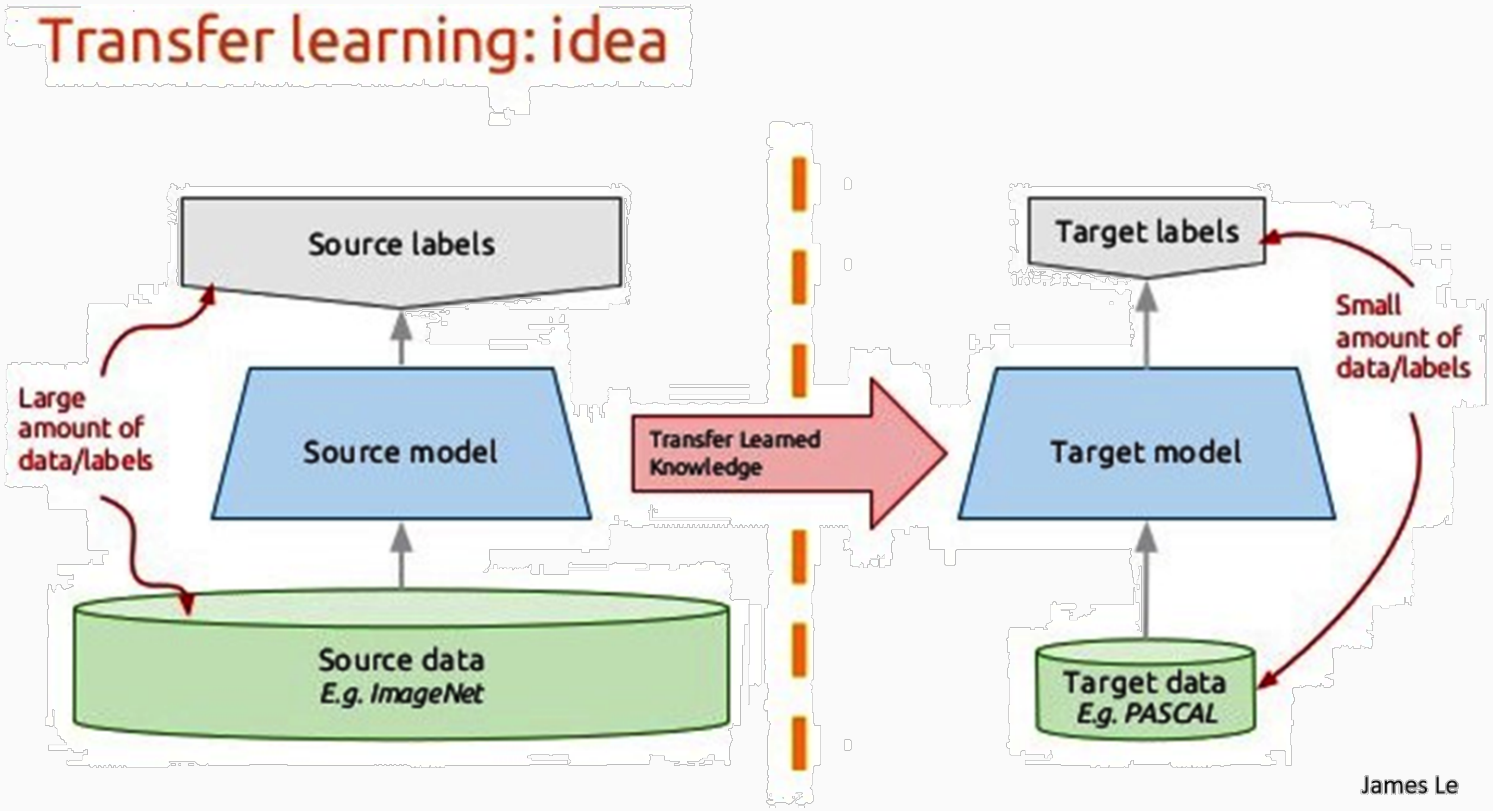


From Annotated to Unsupervised Pre-training

Past and Future Transfer Learning: Before

Supervised Pre-training on large annotated datasets (i.e., ImagetNet)

- Relied on large labeled datasets.
- Limited by the availability of annotated data.
- Example: ResNet

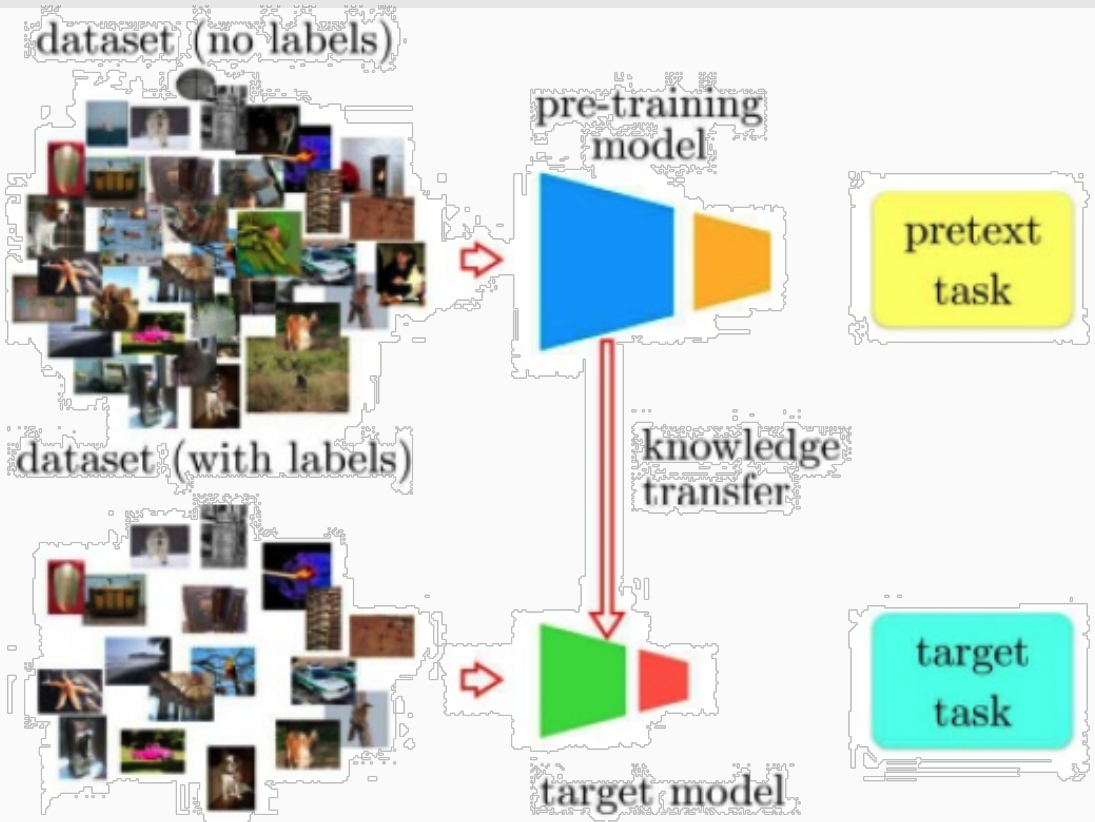


From Annotated to Unsupervised Pre-training

Past and Future Transfer Learning: Now

Unsupervised/Self-supervised Pre-training on Larger unnanotated data

- Utilizes unannotated data, which is abundant.
- Models learn general representations, broader patterns useful for various tasks (e.g., language models predicting the next word).
- Example: GPT-3, LLaMA, SimCLR, BLIP-3



Datasets for Pre-training

Huge Datasets Across Modalities

- **Images:**
  - **ImageNet:** 14 million images, 22,000 categories.
  - **Open Images:** 9 million images with rich annotations.
  - **LAION-5B:** 5 billion image-text pairs.
- **Text:**
  - **Common Crawl:** Petabytes of web text data.
  - **BooksCorpus:** Thousands of books for diverse content.
  - **Wikipedia:** Comprehensive articles across subjects.
- **Audio:**
  - **LibriSpeech:** 1,000 hours of speech.
  - **VoxCeleb:** Speaker recognition dataset with diverse voices.
  - **AudioSet:** Over 2 million labeled audio clips.

- Enable training of large models that capture complex patterns.
- Improve generalization and robustness across tasks.

Pre-training Large Models is the New Norm

Trend

Large-scale models like GPT, BERT, CLIP, Stable Diffusion, trained on vast amounts of data

Massive Computational Power

These models require large computing resources (TPUs/GPUs), but once they are trained they are useful for multiple downstream tasks

Fine-tuning Standard

Instead of training models from scratch, **companies** and researchers fine-tune these models for their own datasets and tasks.

LLMs: Prime Examples of Transfer Learning

- Models of billions of parameters,
- Pre-trained to predict next word
- Unsupervised pre-training on huge corpora of vast and diverse texts
- Learn language patterns by predicting missing words or sentences
- So huge that they are useful for almost everything
- Adjust the model for specific tasks (e.g., translation, summarization, chat)

LLMs: Prime Examples of Transfer Learning

- Models of billions of parameters,
- Pre-trained to predict next word
- Unsupervised pre-training on huge corpora of vast and diverse texts
- Learn language patterns by predicting missing words or sentences
- So huge that they are useful for almost everything
- Adjust the model for specific tasks (e.g., translation, summarization, chat)
- Basic models understand syntax, semantics, and context
- Can be fine-tuned using supervised fine-tuning, reinforcement learning from human feedback (RLHF)

Challenges and Considerations

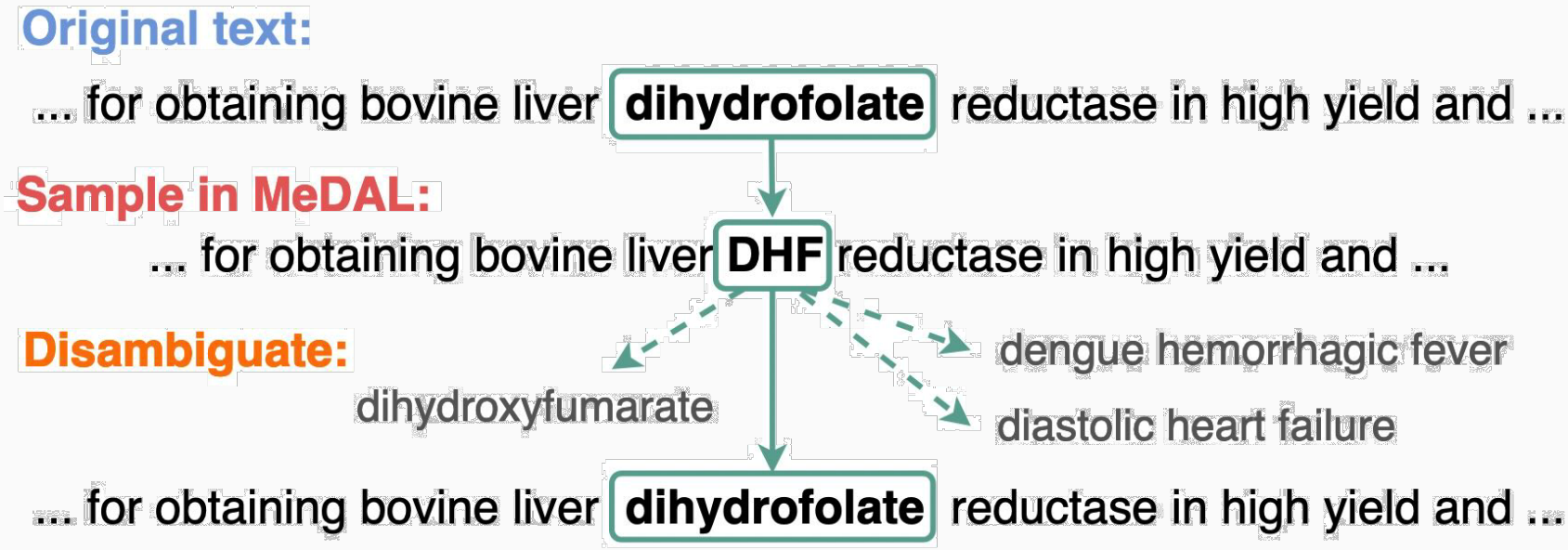
- Computational cost can be insane: 8m of GPU-hours to train LlaMA3, trained over 2.250G USD of GPUs

Challenges and Considerations

- Computational cost can be insane: 8m of GPU-hours to train LlaMA3, trained over 2.250G USD of GPUs
- If the source task is too different from the target domain, performance may degrade because of bad features

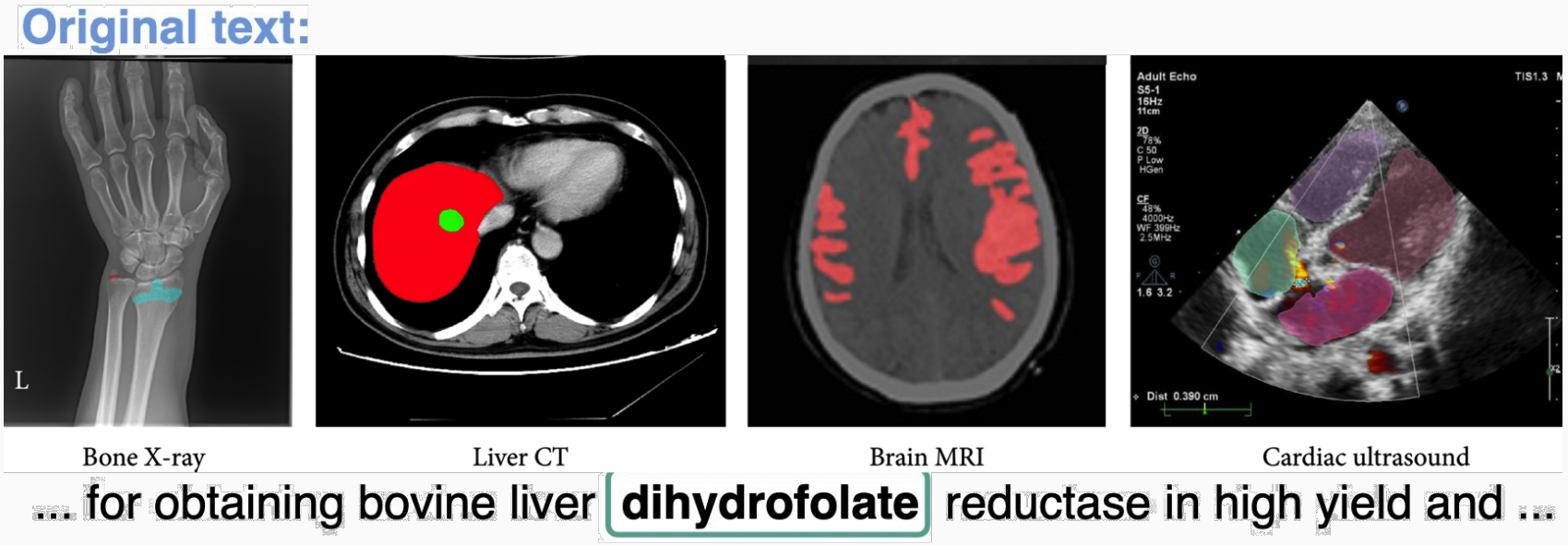
Challenges and Considerations

- Computational cost can be insane: 8m of GPU-hours to train LlaMA3, trained over 2.250G USD of GPUs
- If the source task is too different from the target domain, performance may degrade because of bad features
- If the source domain is too far different from the target domain, performance may degrade (e.g., medical imaging, niche languages)



Challenges and Considerations

- Computational cost can be insane: 8m of GPU-hours to train LlaMA3, trained over 2.250G USD of GPUs
- If the source task is too different from the target domain, performance may degrade because of bad features
- If the source domain is too far different from the target domain, performance may degrade (e.g., medical imaging, niche languages)



Questions?

References i