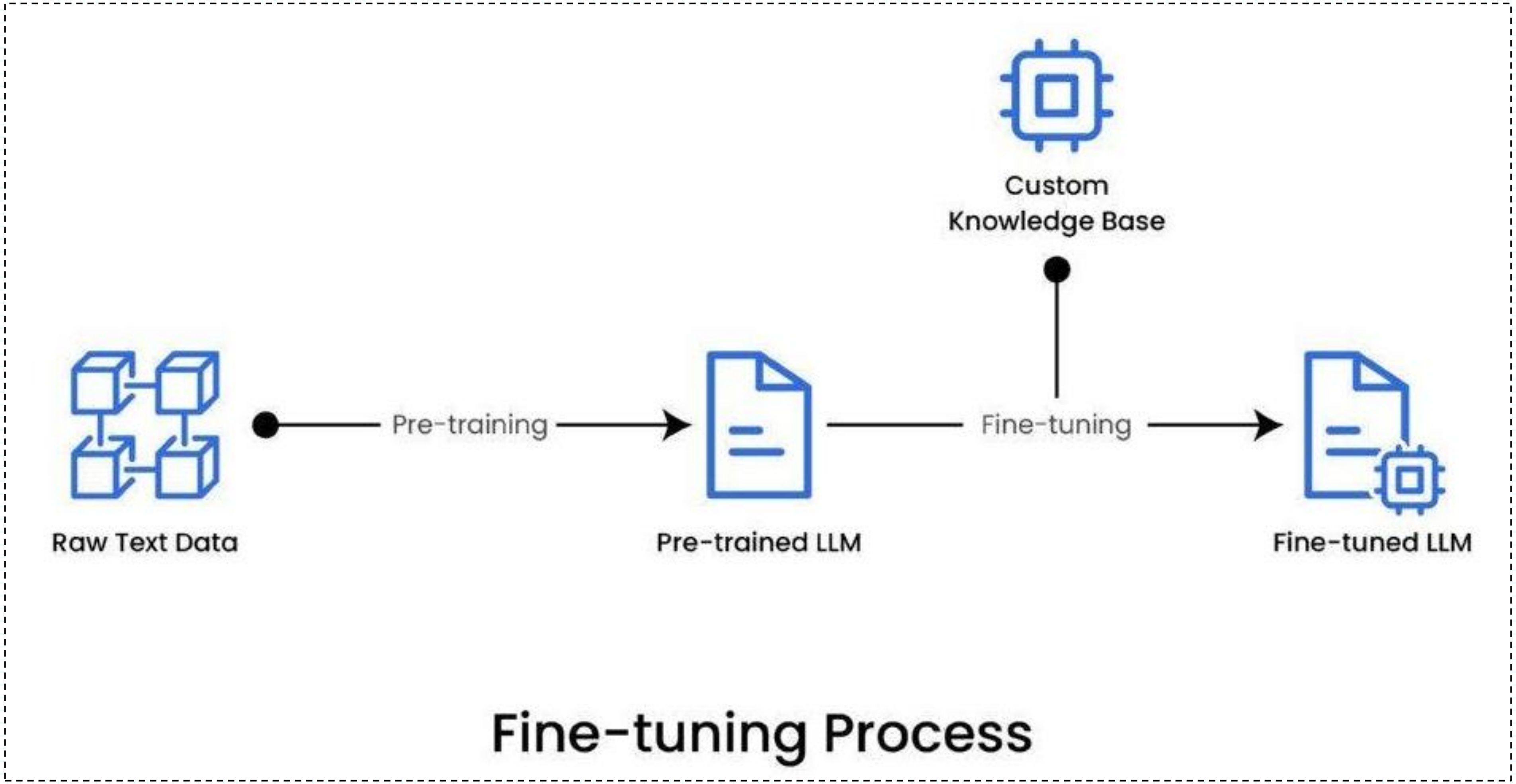
الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

Week 9 | كلية الهندسة | الذكاء الصنعي العملي

Transfer Learning: Fine-Tuning and Domain Adaptation

د. رياض سنبل

What is LLM fine-tuning?



What is LLM fine-tuning?

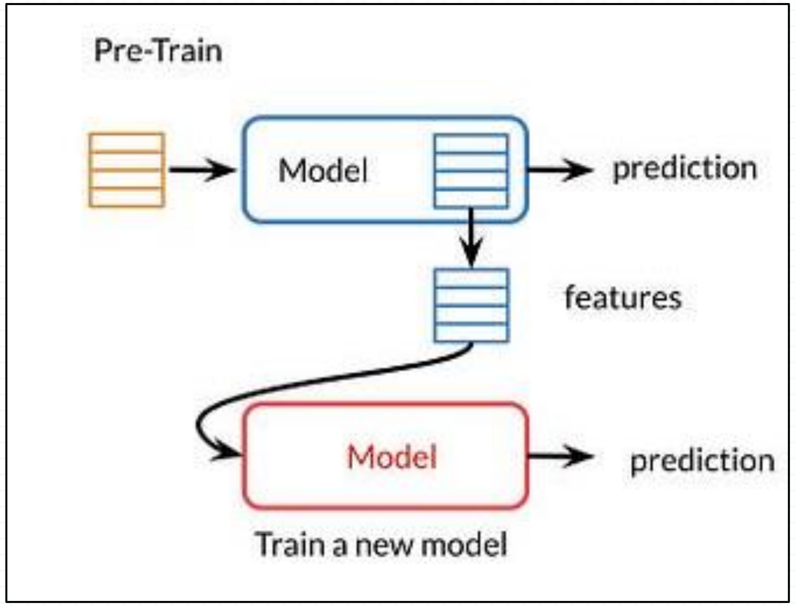
- Fine-tuning is the process of adjusting the parameters of a pre-trained large language model to a **specific task or domain**.
- Although pre-trained language models like GPT possess **vast language knowledge**, **they lack specialization** in specific areas.
- By exposing the model to task-specific examples during fine-tuning, the model can acquire a **deeper understanding of the nuances of the domain**

Why is LLM fine-tuning important?

- **Customization**: Every domain or task has its own unique language patterns, terminologies, and contextual nuances.
- **Data compliance**: In many industries, such as healthcare, finance, and law, strict regulations govern the use and handling of sensitive information. Organizations can ensure their model adheres to data compliance standards by fine-tuning the LLM on proprietary or regulated data.
- **Limited labeled data:** Fine-tuning allows organizations to leverage pre-existing labeled data more effectively by adapting a pre-trained LLM to the available labeled dataset, maximizing its utility and performance.

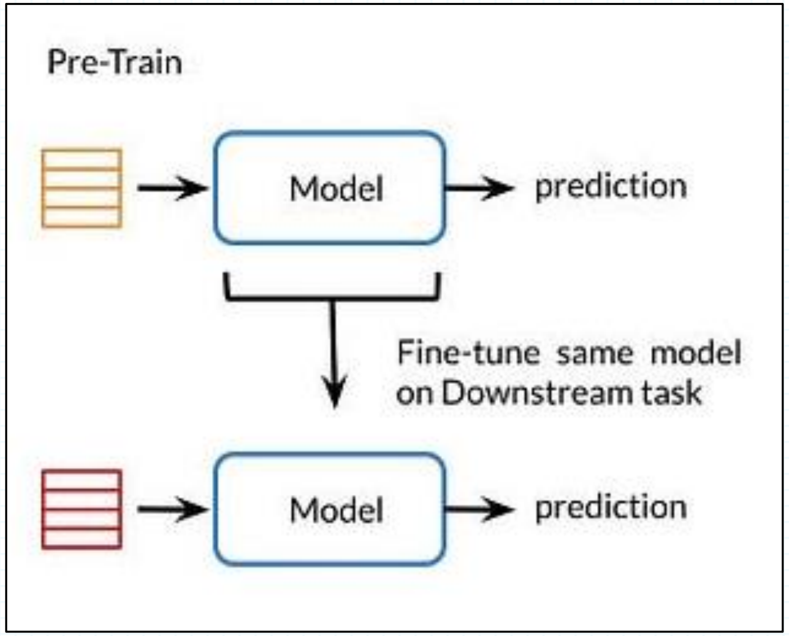
Types of LLM fine-tuning

- **Feature extraction:**
  - It involves freezing the weights of a pre-trained model's layers and using it as a feature extractor
  - Used when a smaller dataset is available, and the target domain is closely aligned with the original domain of the pre-trained model.
  - Faster training, requires less computational resources, and can be more effective when the new dataset is small.



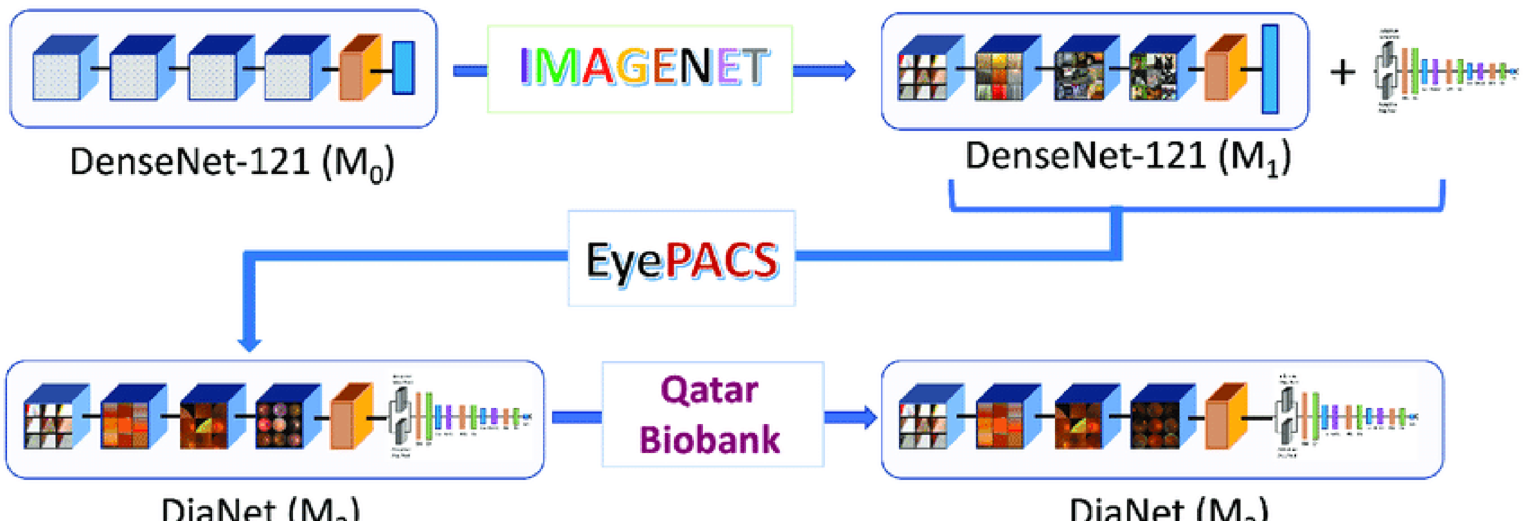
Types of LLM fine-tuning

- **Full fine-tuning**
  - involves unfreezing some or all of the pre-trained layers and retraining them along with new layers.
  - Used when a larger dataset is available and more flexibility is needed to adapt the pre-trained model to the specific task
  - Allowing the model to learn new features that are specific to the target task
  - Can achieve higher accuracy, better adaptability to the new task, and allows the model to learn more task-specific features while preserving the general knowledge from the original dataset.



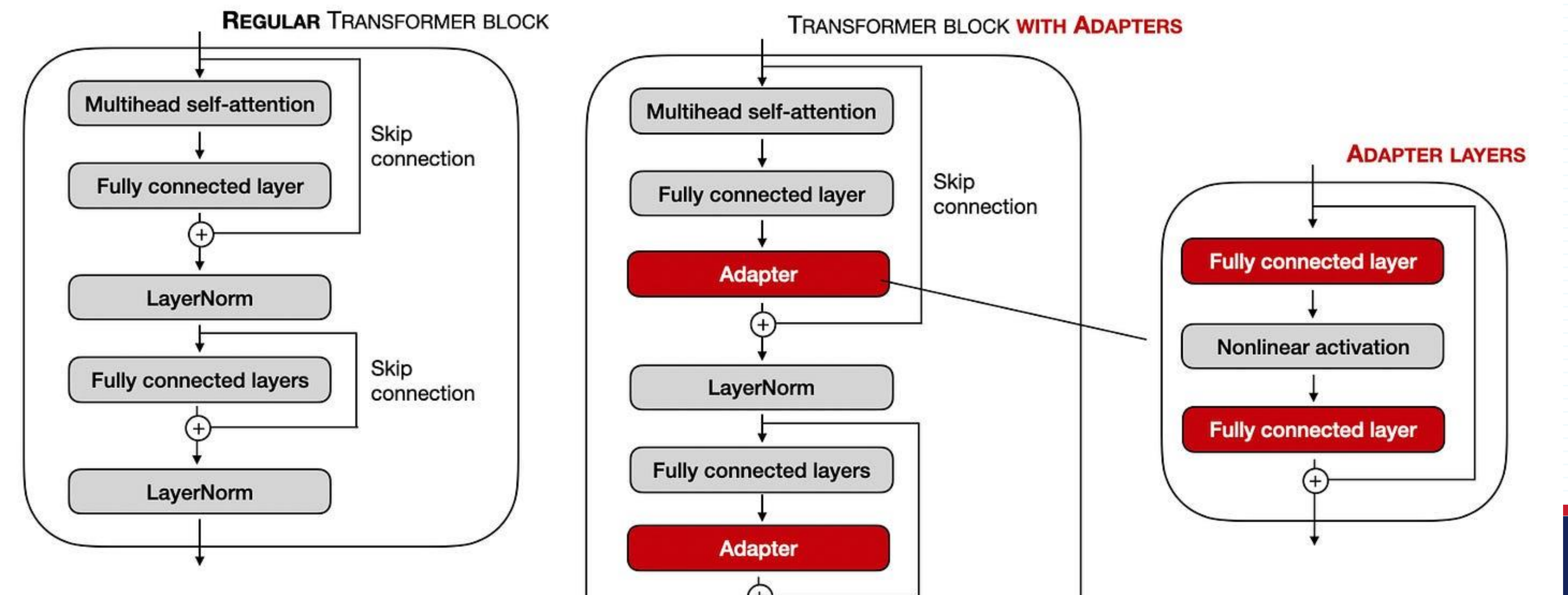
Types of LLM fine-tuning

- **Multi-stage fine-tuning**
  - A technique where a model is trained in multiple sequential stages, each stage building upon the previous one.
  - The process might involve:
    - Pre-training
    - Initial Fine-Tuning: smaller dataset relevant to a specific domain e.g. medical text.
    - Domain-Specific Fine-Tuning: further fine-tuned on a dataset specifically related to a narrow sub-domain, such as clinical trial reports.



Types of LLM fine-tuning

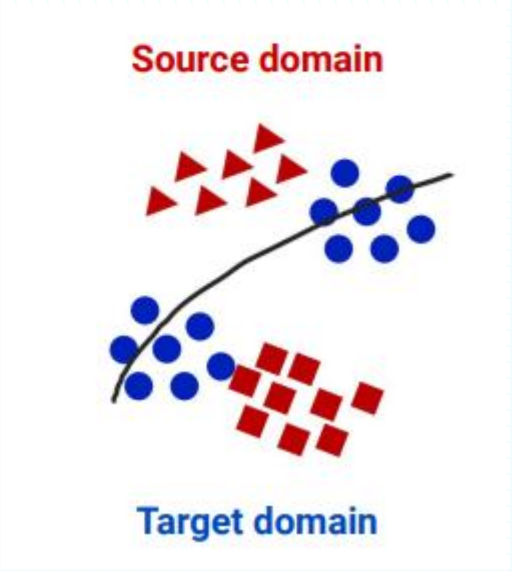
- **Adapter fine-tuning** is a parameter-efficient technique that allows for efficient adaptation of large language models (LLMs) to specific tasks by training small, task-specific "adapter" modules instead of updating all model parameters.
- Adapters: Adapters are small neural network layers (typically a pair of fully connected layers) inserted into the pre-trained model.

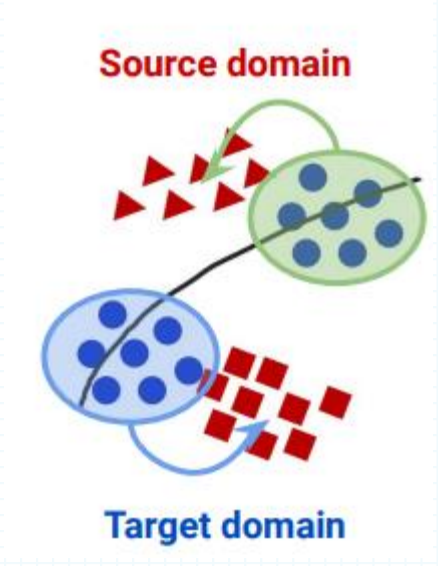


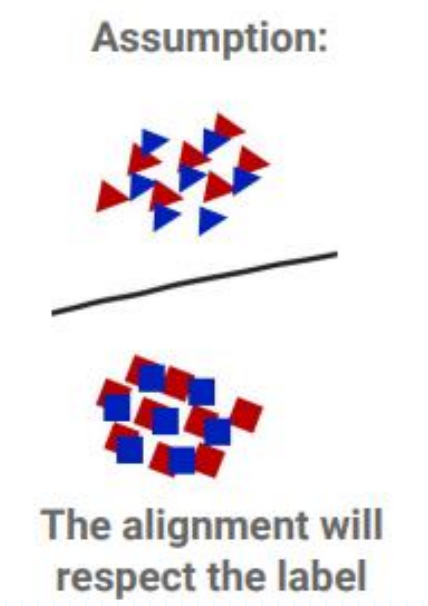
Domain Adaptation

- Domain Adaptation (DA) is a machine learning technique to transfer knowledge from a **"source" domain** (with lots of labeled data) to a different but related **"target" domain** (often with little or no labeled data)
- Unsupervised domain adaptation (UDA) **aims to learn a predictive model for an unlabeled domain** by transferring knowledge from a separate labeled source domain.

Unsupervised Domain Adaptation

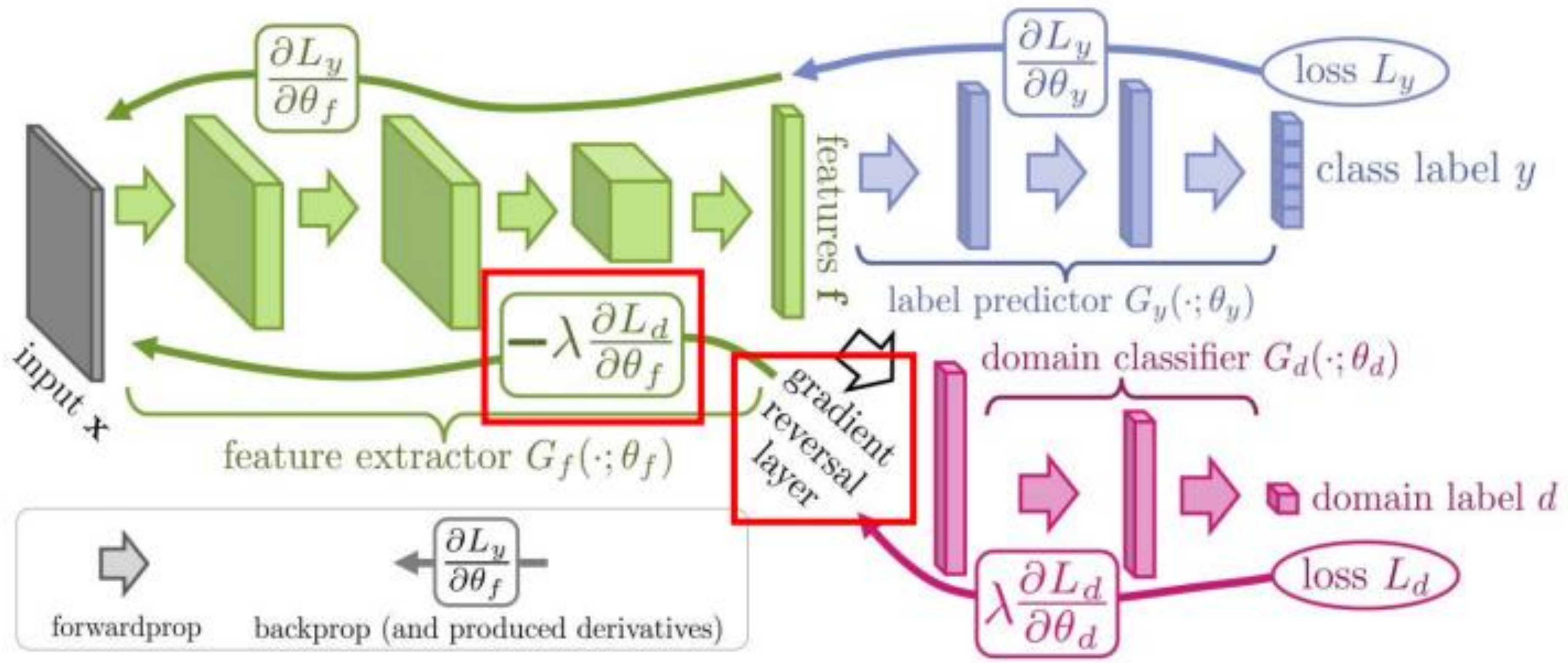






Unsupervised Domain Adaptation

$$L_{\text{total}}(\theta_f) = L_y(\theta_f, \theta_y) \; - \; \lambda \, L_d(\theta_f, \theta_d)$$



Unsupervised Domain Adaptation

Method	MNIST → USPS	USPS → MNIST	SVHN → MNIST
Source only	0.752 ± 0.016	0.571 ± 0.017	0.601 ± 0.011
Gradient reversal	0.771 ± 0.018	0.730 ± 0.020	0.739 [16]
Domain confusion	0.791 ± 0.005	0.665 ± 0.033	0.681 ± 0.003
CoGAN	0.912 ± 0.008	0.891 ± 0.008	did not converge
ADDA (Ours)	0.894 ± 0.002	0.901 ± 0.008	0.760 ± 0.018