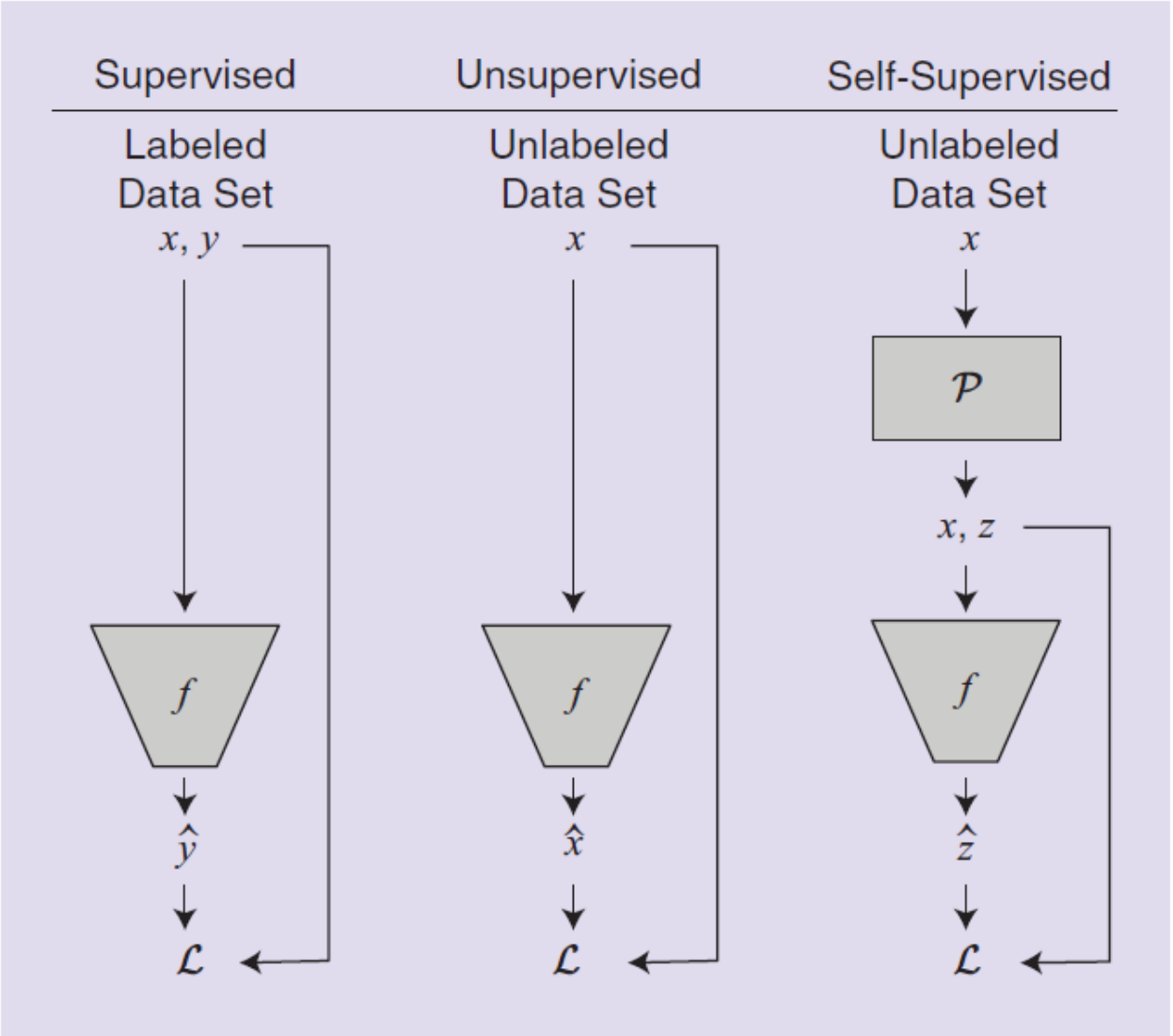


FIGURE 1. Contrasting supervised, unsupervised and self-supervised learning paradigms for training a model f using raw data x, labels y, and loss function $\mathcal{L}$. Self-supervision methods introduce pretext tasks $\mathcal{P}$ that generate pseudolabels z for discriminative training of f.

Module – 2

Basics of Foundation Models, SSL formalisms, definitions and examples of

- **Pretext tasks**
- **Losses**
- **Downstream adaptations**

in a domain-agnostic setting

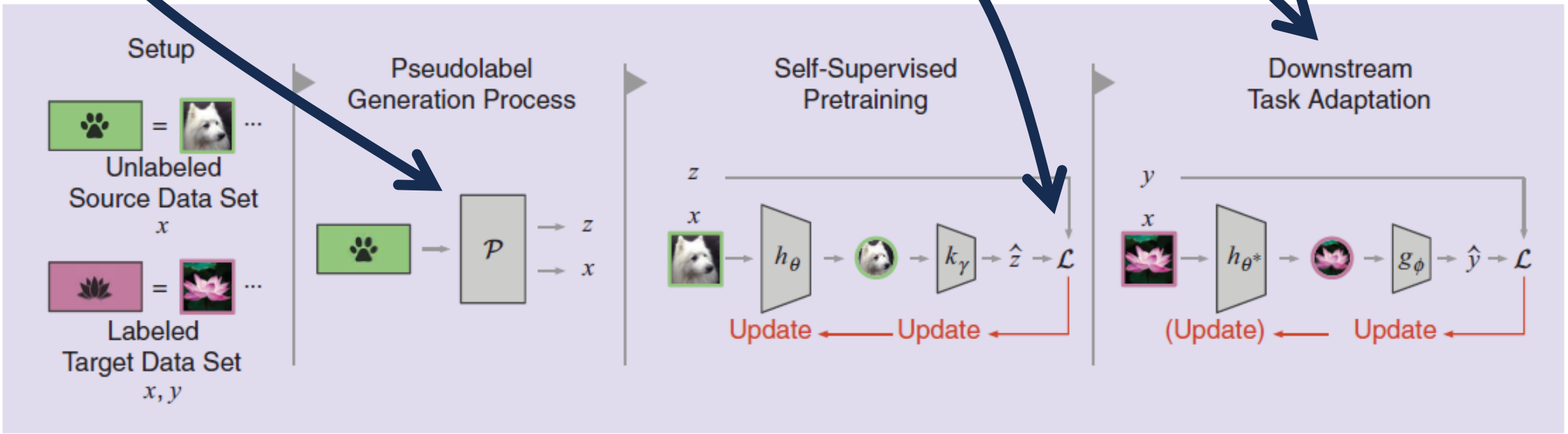


FIGURE 2. The self-supervised workflow starts with an unlabeled source data set and a labeled target data set. As defined by the pretext task, pseudolabels are programmatically generated from the unlabeled set. The resulting inputs, x and pseudolabels z, are used to pretrain the model $k_\gamma(h_\theta(\cdot))$—composed of feature extractor h_θ and output k_γ modules—to solve the pretext task. After pretraining is complete, the learned weights θ^* of the feature extractor h_{θ^*} are transferred and used together with a new output module g_ϕ to solve the downstream target task.

4.10 Theory

Authors: Aditi Raghunathan, Sang Michael Xie, Ananya Kumar, Niladri Chatterji, Rohan Taori, Tatsunori Hashimoto, Tengyu Ma

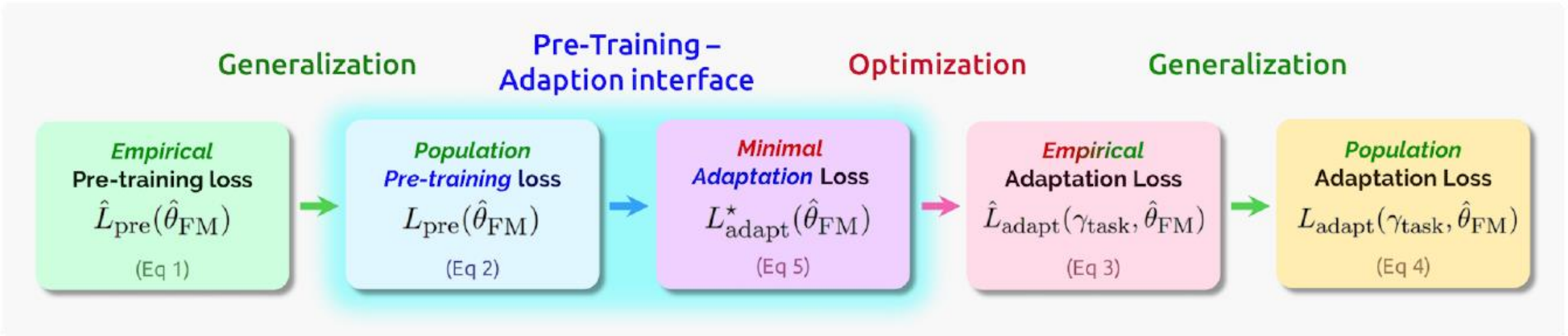
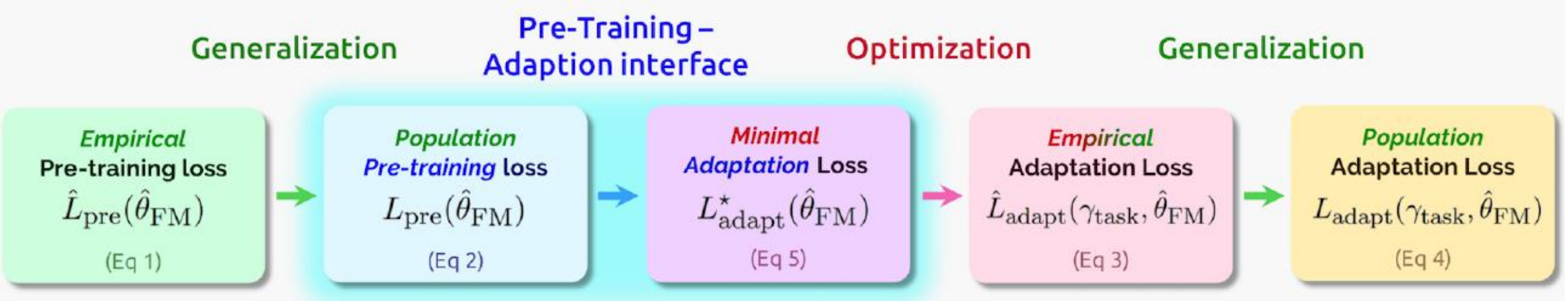


Fig. 22. The analysis of foundation models from pretraining on diverse data to downstream performance on adapted tasks involves capturing the relation between different loss terms as shown above. The main challenge is to analyze the highlighted pretraining-adaptation interface which requires reasoning carefully about the population losses in addition to the model architecture, losses and data distributions of the pretraining and adaptation stages (§4.10.2: THEORY-INTERFACE). Analysis of generalization and optimization largely reduces to their analysis in standard supervised learning.



As shown in Figure 22, the main missing link beyond standard supervised theory is:

Under what conditions does a small population pretraining loss $L_{pre}(\hat{\theta}_{\text{FM}})$ *imply a small minimal adaptation loss* $L^{\star}_{adapt}(\hat{\theta}_{\text{FM}})$ and why?

Supervised Learning

Target task: t

Supervised (Labeled) Data

$$D_t = \{x_i^{(t)}, y_i^{(t)}\}_{i=1}^{N}$$

N sample training data.

For each x ($x_i^{(t)}$) $\in D_t$

predicted label: $f(x) = \hat{y}$

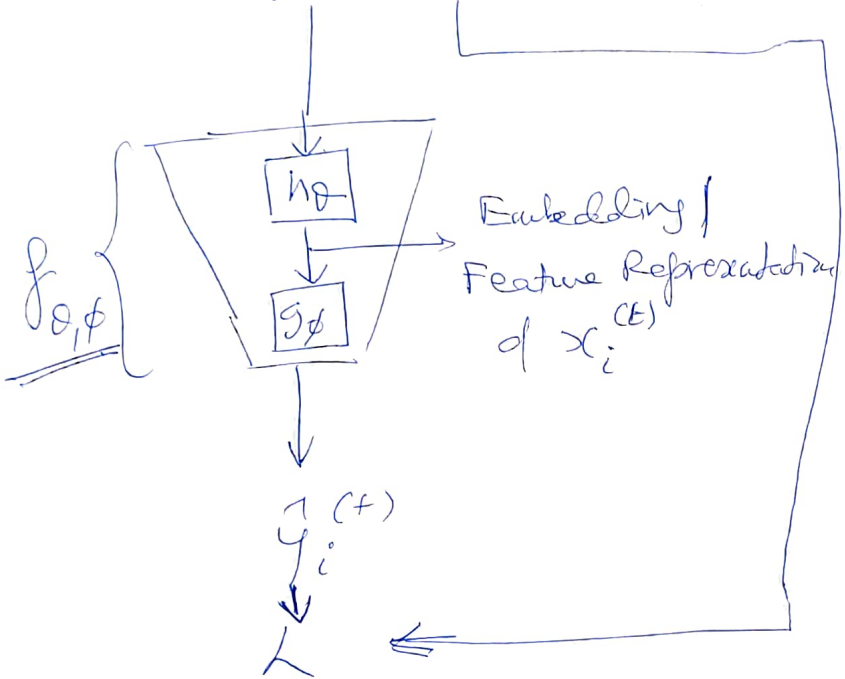
$$f \rightarrow g_\phi(h_\theta)$$

h_θ : Representation feature extractor fn.

g_ϕ : classifier/regression fn.

$$\hat{y}_i^{(t)} = g_\phi(h_\theta(x_i^{(t)}))$$

$D_t = \{x_i^{(t)}, y_i^{(t)}\}_{i=1}^{N}$



$L(\hat{y}_i^{(t)}, y_i^{(t)})$

$L(f(x_i^{(t)}), y_i^{(t)})$

$L(g_\phi(h_\theta(x_i^{(t)})), y_i^{(t)})$

$\Rightarrow L(i, \theta, \phi)$

Total loss on D_t

[Full Batch Representation]

$$L(\theta, \phi) = \sum_{i=1}^{N} L(i, \theta, \phi)$$

$$f^* = \theta^*, \phi^*$$

$$= \underset{\theta,\phi}{\text{argmin}}\ L(\theta, \phi)$$

$$f^* = \theta^*, \phi^* = \underset{\theta,\phi}{\operatorname{argmin}} \mathcal{L}(\theta, \phi)$$

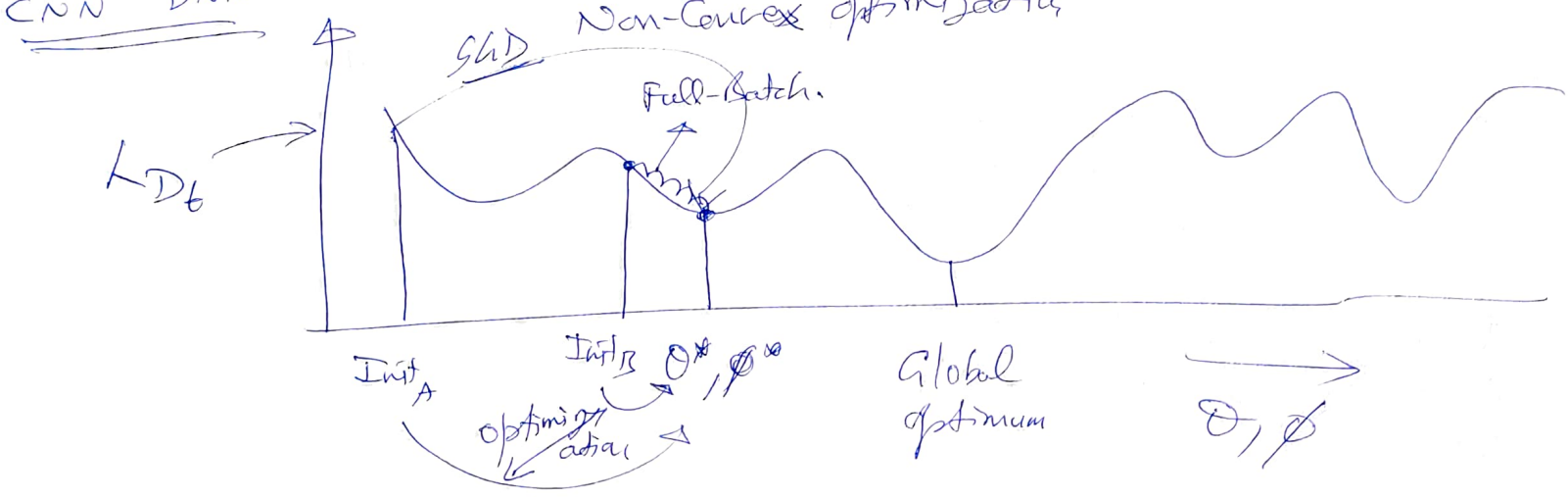
$$= \underset{\theta,\phi}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}\left(\underbrace{g_\phi(h_\theta(x_i^{(+)}))}_{\hat{y}_i^{(+)}}, y_i^{(+)}\right)$$

$h_\theta, g_\phi \Rightarrow \theta, \phi$: 100 — 1000s of millions of parameters

: Deep backbones

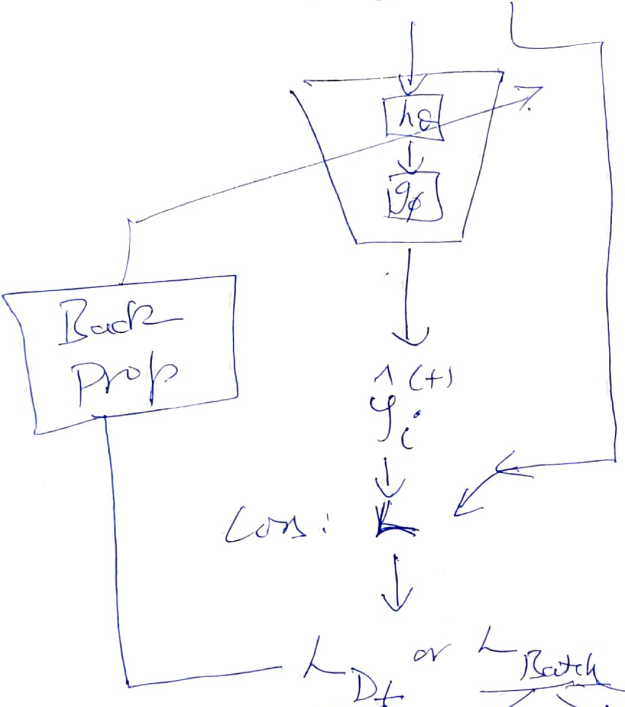
CNN DNN

$$\mathcal{L}_{D_t} = \sum_{x,y \in D_t} \mathcal{L}(\cdot,\cdot)$$



finding θ^*, ϕ^* : $f^* \rightsquigarrow h_{\theta^*}, g_{\phi^*}$

$D_t = \{x_i^{(t)}, y_i^{(t)}\}_{i=1}^{N_t}$



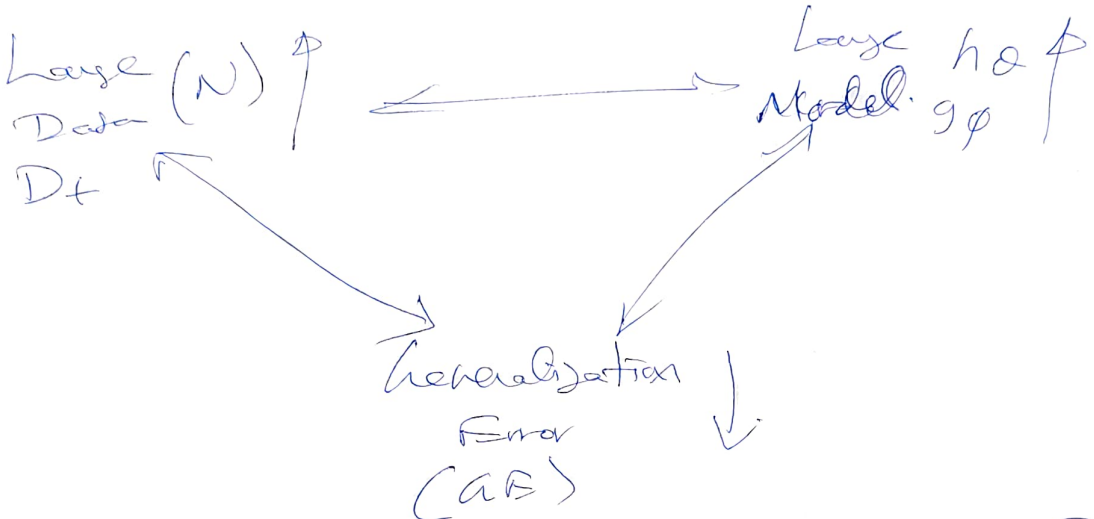
optimization

By Gradient Descent

→ Back Propagation

→ Updated wts using

Ins fn. L_{Batch}



1. Given unseen test/inference → Larger Dt [For Low GE]
   Larger Dt → More Complex Decision Surfaces
   → Complex Models

2. Given a task → UAT → Large/Deep Models (More Complex Models)
   By PAC/Sample Complexity (SC) Bound → Larger Dt (N > SC)

Unsupervised Learning

- Learn from 'unlabeled' data
  - Learn Compact 'Latent' representations
  - Build Generative Models
  - Or Density Estimators

Gaussian Mixtures

clustering

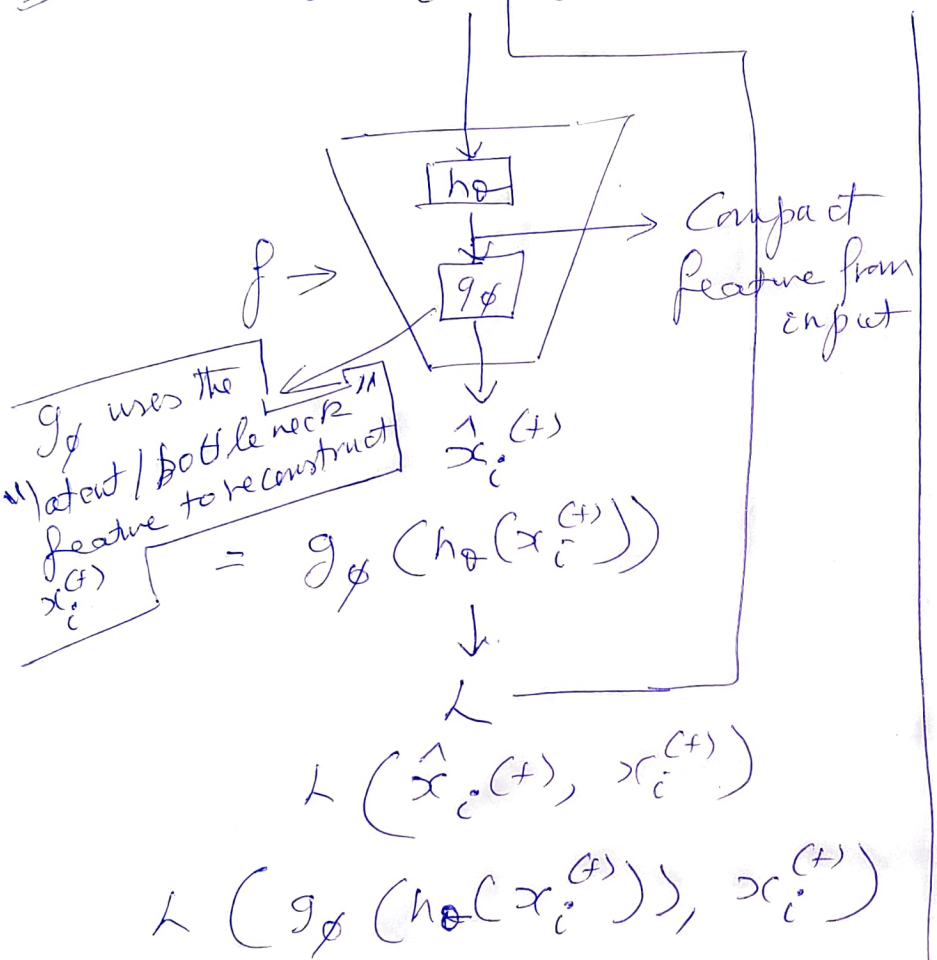
AE (Auto Encoder)

VAEs

GANs

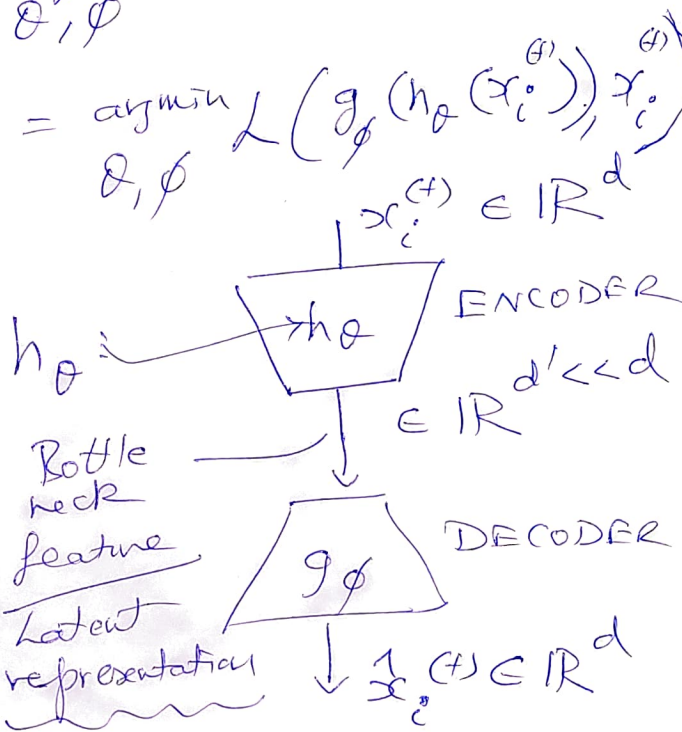
AUTOENCODER

$D_t = \{x_i^{(t)}\}_{i=1}^{N}$: No label — $y_i^{(t)}$



AEs optimise θ, ϕ on "reconstruction" objective

$\theta^*, \phi^* = \underset{\theta,\phi}{\operatorname{argmin}} \mathcal{L}(g_\phi(h_\theta(x_i^{(t)})), x_i^{(t)})$



PRETRAINING Self-Supervised Learning DOWNSTREAM

Unlabeled $D_S = \{x_i^{(s)}\}_{i=1}^{M}$ $M \gg N$

Pretext Task Process: P

$\bar{D}_S = P(D_S) = \{x_i, z_i\}_{i=1}^{M}$

Pretext Model

h_θ Feature extractor

R_ϕ Output Modules

update θ, ϕ

$\hat{z}_i$

$\mathcal{L}$

Discarded

Labeled $D_t = \{x_i^{(t)}, y_i^{(t)}\}_{i=1}^{N}$

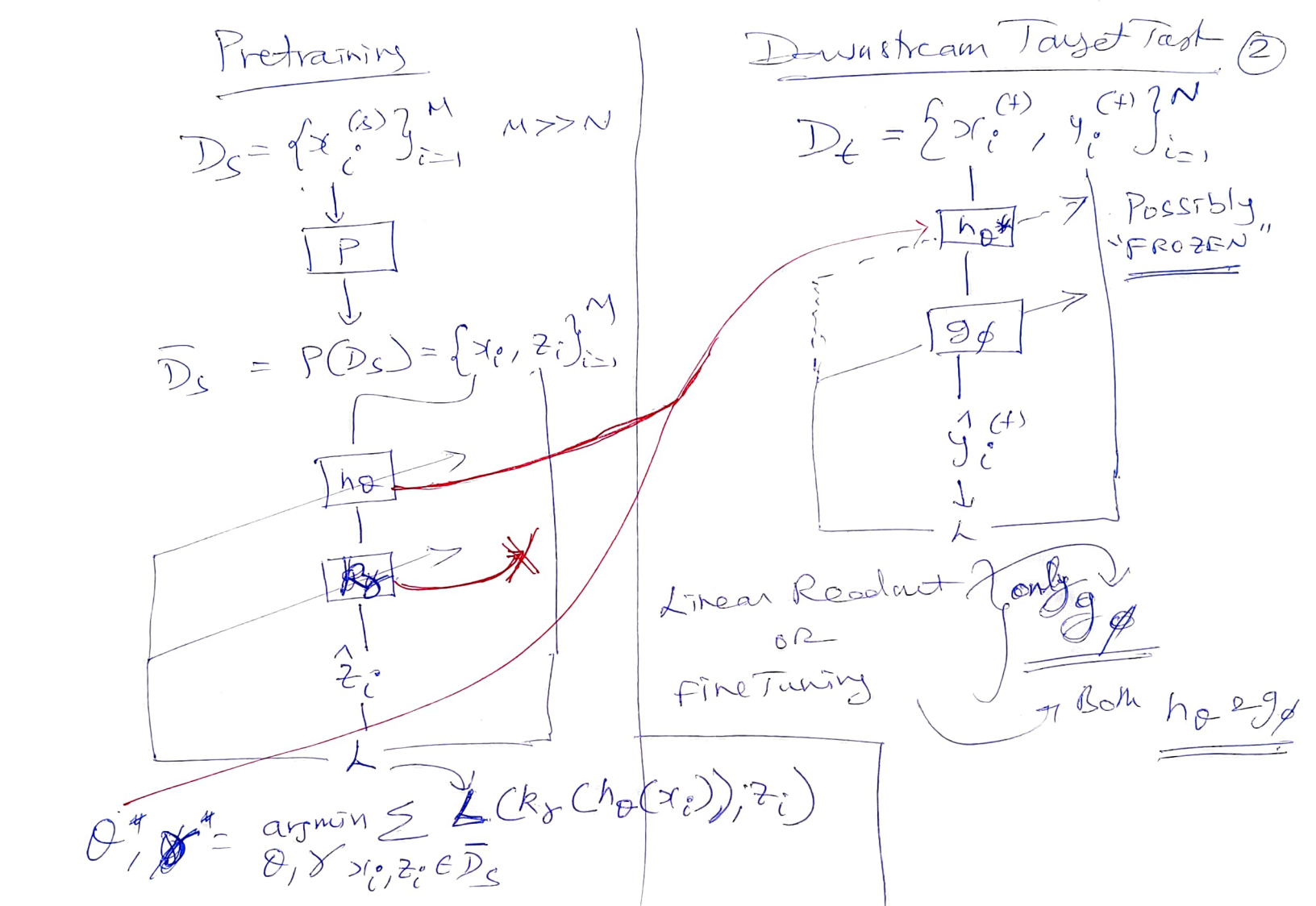
h_θ^* Feature extractor

Feature Repn.

g_ϕ → classifier

$\hat{y}_i^{(t)}$

$\mathcal{L}$


Pretraining
$D_S = \{x_i^{(s)}\}_{i=1}^{M}$ $M >> N$
P
$\bar{D}_S = P(D_S) = \{x_i, z_i\}_{i=1}^{M}$
h_θ
g_γ
$\hat{z}_i$
$\mathcal{L}$
$\theta^*, \gamma^* = \underset{\theta,\gamma}{\operatorname{argmin}} \sum_{x_i, z_i \in \bar{D}_S} \mathcal{L}(k_\gamma(h_\theta(x_i)), z_i)$
Downstream Target Task (2)
$D_t = \{x_i^{(t)}, y_i^{(t)}\}_{i=1}^{N}$
h_{θ^*}
Possibly "FROZEN"
g_ϕ
$y_i^{(t)}$
$\mathcal{L}$
Linear Readout or FineTuning
only g_ϕ
Both h_θ & g_ϕ

Linear Readout / Linear Probing

(θ, δ) : Pretrained model weights / Parameters

- h_θ — Feature extractor
- R_δ — Pretext task "specific" head.

Downstream task :

- Reuse h_θ for new task
- Replace R_δ with a new head g_ϕ designed for the new task.

h_θ : Frozen

g_ϕ : Trained / optimized →

$$\phi^* = \arg\min_{\phi} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(g_\phi(h_\theta(x_i^{(t)})), y_i^{(t)}\right)$$

Fine-tuning

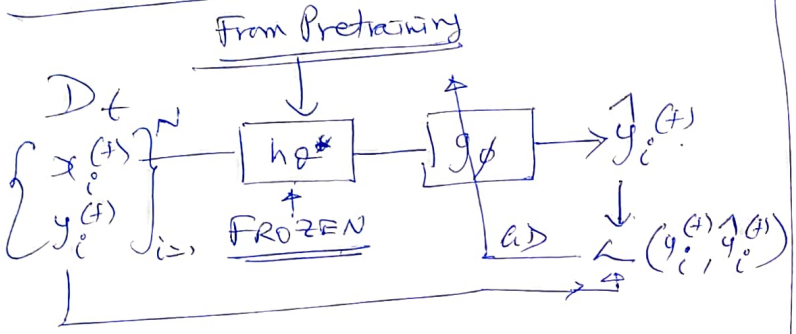
- Retrain the entire network

$h_{\theta^*} + g_\phi$

$$\theta^{**}, \phi^* = \underset{\theta,\phi}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(g_\phi(h_\theta(x_i^{(t)})), y_i^{(t)}\right)$$

Initialize with θ^*

Linear Readout / Probing

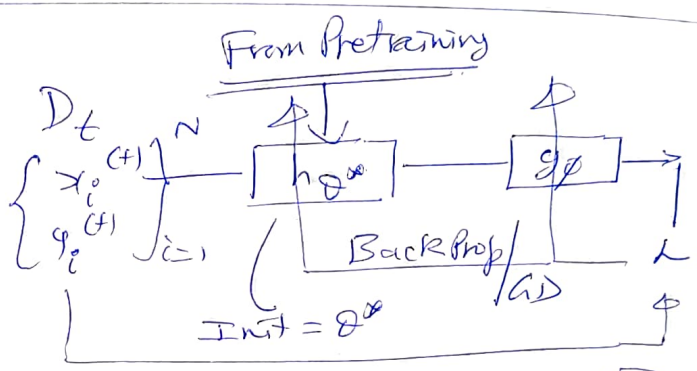


- $N \Rightarrow$ Typically "small" $\Rightarrow$ D_t : Labeled Dataset of Target Task

$N \Rightarrow$ Very Small } Sparse Data regime

- # unique parameters to learn for the target task must be aggressively limited to avoid overfitting.
- g_ϕ : Simple Linear fn || 1-layer FC

Finetuning



$N \Rightarrow$ "Moderately more data"

- Enough target data to refine all the model parameters
- OR when Pretext task / data NOT suited to downstream task.

SSL Scenarios

- $N \uparrow$: N is large (Dense labels are available for the target task.)
  D_t is large
  → Direct "Supervised learning"
  — Best approach.
  — SSL may not be helpful

- Domain (Pretraining) ≠ Domain (Target)
  Very different
  ↳ Indomain

Indomain SSL pretraining

eg Imagenet ~~~~→ Radar imaging

SSL: Not very effective

BUT Pretrain on Unlabeled Data with SSL

~→ Highly effective.

SSL Scenarios (Contd.)

- Domain (Pre Training) $\approx$ Domain (Target)

Domain (Pre Training) → Imagenet

Domain (Target) → everyday objects

SSL — Highly effective.

General Transfer Learning

Supervised Pretraining

vs

Unsupervised Pretraining (SSL)

→ Superior

SSL Design Choices

- h_θ + Frozen, g_ϕ trained (L.R) — when?
- $[h_\theta^{*(init)} + g_\phi]$ trained (F.T)

when?
- $D_S \approx D_T$?
- $\|D_\epsilon\|$ Academia
- Performance Comparisons.

- $[h_\theta, k_\gamma]$ split in a
  - Pretraining Backbone.
  - h_θ: early layers } encode "task/data agnostic" patterns
    : good for "domain mismatch" Conditions
- Pretraining — Stopping Criterion [Validation ?]
- Downstream Performance $\propto$ Pretrains Model Complexity