

# Decision-focused Learning in Partial Index Tracking



**Hyunglip Bae**

Ph.D, KAIST

[qogudflq@kaist.ac.kr](mailto:qogudflq@kaist.ac.kr)

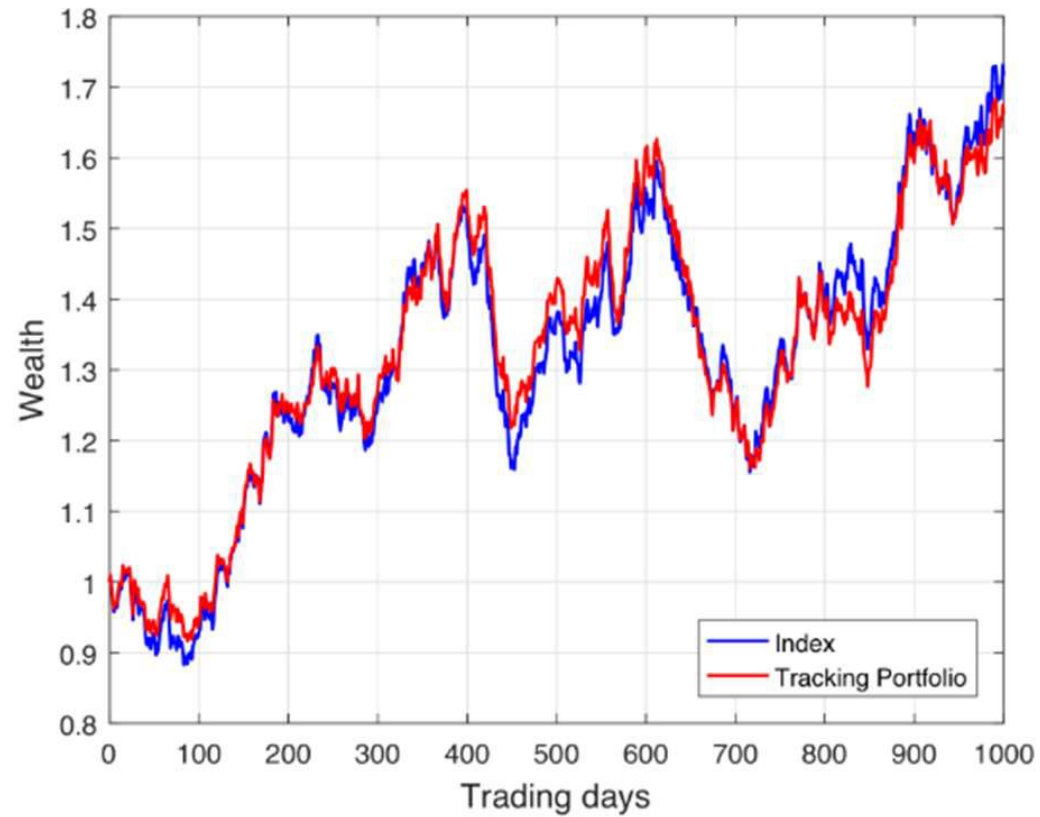
Financial Engineering Lab

Department of Industrial Engineering



# Introduction

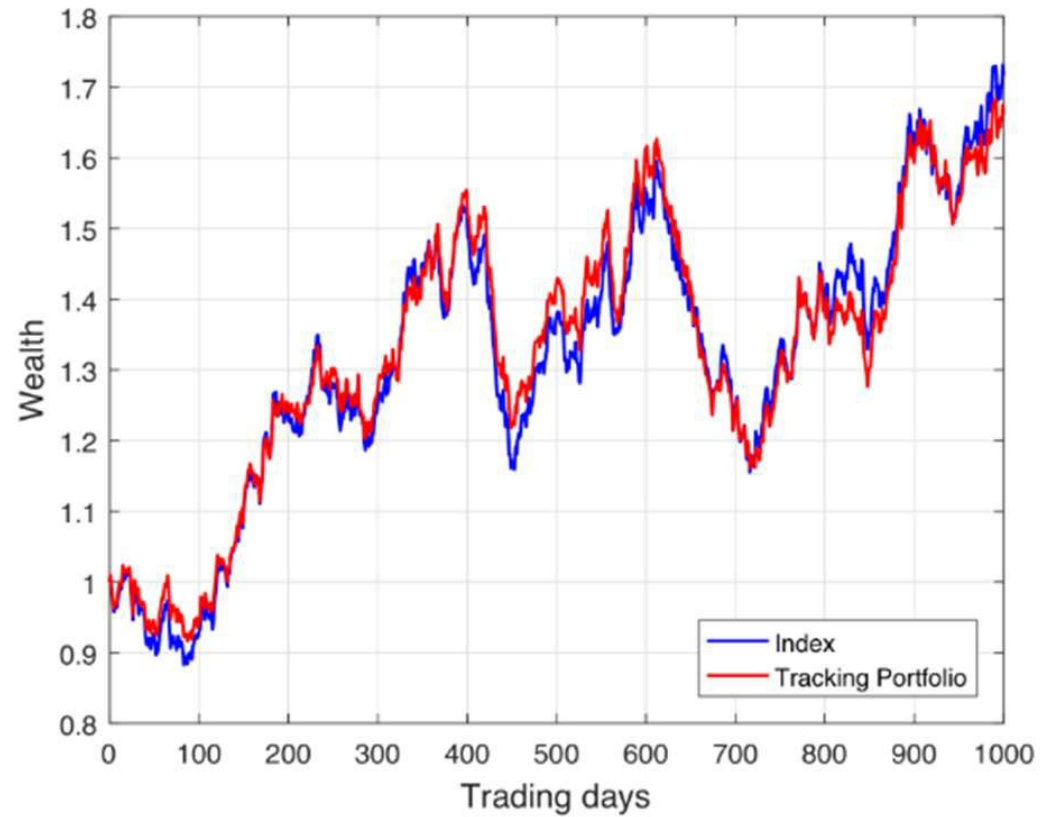
- **Partial Index Tracking**



- **Index:** measure the performance of finance market

# Introduction

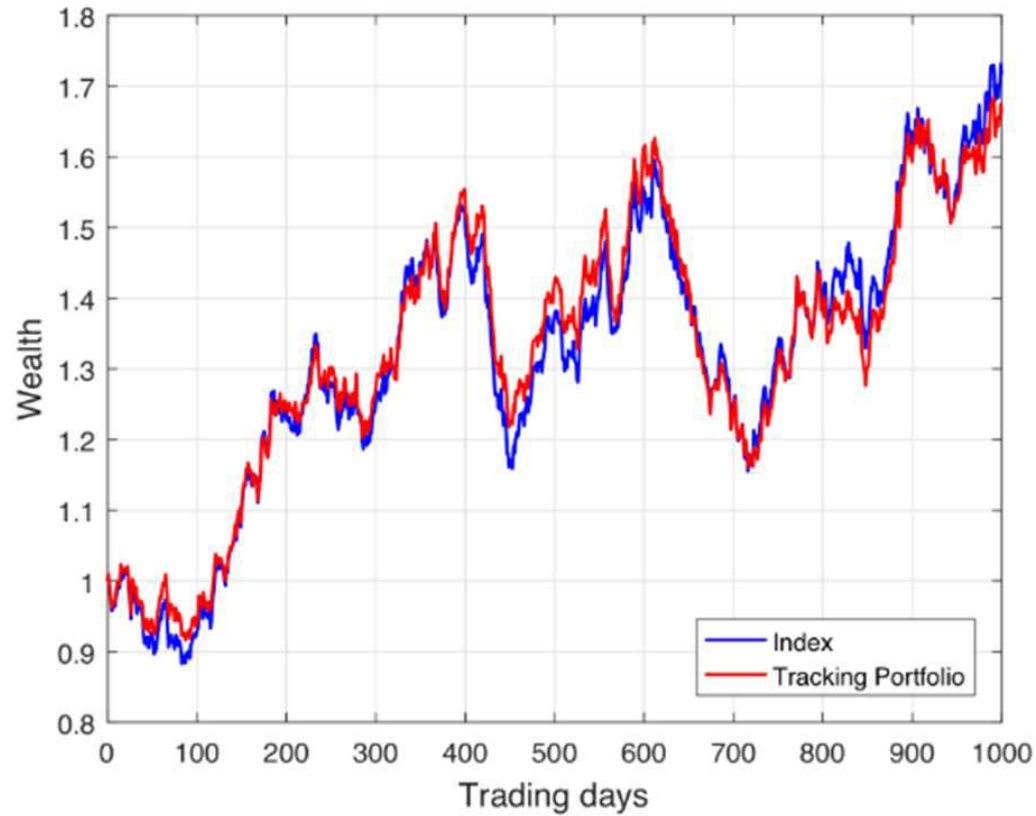
- **Partial Index Tracking**



- **Index:** measure the performance of finance market
- **Tracking:** aim to replicate performance of index

# Introduction

- **Partial Index Tracking**



- **Partial:** use only subset of assets in index
- **Index:** measure the performance of finance market
- **Tracking:** aim to replicate performance of index

## Basic Formulation

- Variables & Functions

Name	Definition
$n \in \mathbb{N}$	The number of assets
$\hat{r} \in \mathbb{R}^n$	The return of assets ( <b>Uncertain</b> )
$w \in \mathbb{R}^n$	The weight of portfolio
$I \in \mathbb{R}$	Target Index
<b>1</b>	One vector

Name	Definition
<b>Card(<math>\cdot</math>)</b>	The number of positive component
$\ \cdot\ _2$	Euclidean 2-norm
$\ \cdot\ _1$	Euclidean 1-norm

## Basic Formulation

- Variables & Functions

Name	Definition
$n \in \mathbb{N}$	The number of assets
$\hat{r} \in \mathbb{R}^n$	The return of assets ( <b>Uncertain</b> )
$w \in \mathbb{R}^n$	The weight of portfolio
$I \in \mathbb{R}$	Target Index
$\mathbf{1}$	One vector

Name	Definition
<b>Card</b> ( $\cdot$ )	The number of positive component
$\ \cdot\ _2$	Euclidean 2-norm
$\ \cdot\ _1$	Euclidean 1-norm

- Formulation

$$\text{Minimize}_w \quad \|\hat{r}^\top w - I\|^2 \quad \Rightarrow \text{Minimize tracking error}$$

# Basic Formulation

- Variables & Functions

Name	Definition
$n \in \mathbb{N}$	The number of assets
$\hat{r} \in \mathbb{R}^n$	The return of assets ( <b>Uncertain</b> )
$w \in \mathbb{R}^n$	The weight of portfolio
$I \in \mathbb{R}$	Target Index
$\mathbf{1}$	One vector

Name	Definition
<b>Card</b> ( $\cdot$ )	The number of positive component
$\ \cdot\ _2$	Euclidean 2-norm
$\ \cdot\ _1$	Euclidean 1-norm

- Formulation

$$\begin{array}{lll}
 \text{Minimize}_w & \|\hat{r}^\top w - I\|^2 & \Rightarrow \text{Minimize tracking error} \\
 \text{s.t} & \mathbf{1}^\top w = 1 & \Rightarrow \text{Sum of portfolio weight} = 1
 \end{array}$$

# Basic Formulation

## • Variables & Functions

Name	Definition
$n \in \mathbb{N}$	The number of assets
$\hat{r} \in \mathbb{R}^n$	The return of assets ( <b>Uncertain</b> )
$w \in \mathbb{R}^n$	The weight of portfolio
$I \in \mathbb{R}$	Target Index
$\mathbf{1}$	One vector

Name	Definition
<b>Card</b> ( $\cdot$ )	The number of positive component
$\ \cdot\ _2$	Euclidean 2-norm
$\ \cdot\ _1$	Euclidean 1-norm

## • Formulation

$$\begin{aligned}
 & \underset{s.t.}{\text{Minimize}_w} && \|\hat{r}^\top w - I\|^2 && \Rightarrow \text{Minimize tracking error} \\
 & && \mathbf{1}^\top w = 1 && \Rightarrow \text{Sum of portfolio weight} = 1 \\
 & && \text{Card}(w) \leq k && \Rightarrow \text{Cardinality constraint}
 \end{aligned}$$



## **DFL in partial index tracking**

- **Challenge**
  - Cardinality constraint makes the problem discrete and not differentiable

## DFL in partial index tracking

- **Challenge**

- Cardinality constraint makes the problem discrete and not differentiable
- In this presentation, we introduce two strategies for this challenge

### 1) Semi-definite Relaxation Approach

$\min_{W \in S_n^+}$	$\mathbf{Tr}(\hat{R}W) - 2(\mathbf{1}^\top W \hat{R}C)$
	$\mathbf{Tr}(I_N W) = 1$
	$\mathbf{1}^\top  W  \mathbf{1} \leq k \mathbf{Tr}(W)$
	$W \in S_n^+$

# DFL in partial index tracking

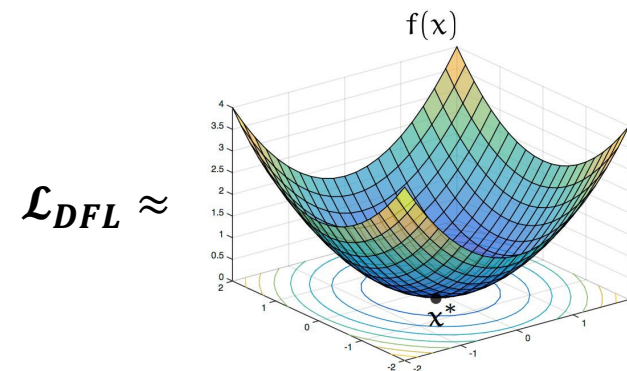
- Challenge

- Cardinality constraint makes the problem discrete and not differentiable
- In this presentation, we introduce two strategies for this challenge

## 1) Semi-definite Relaxation Approach

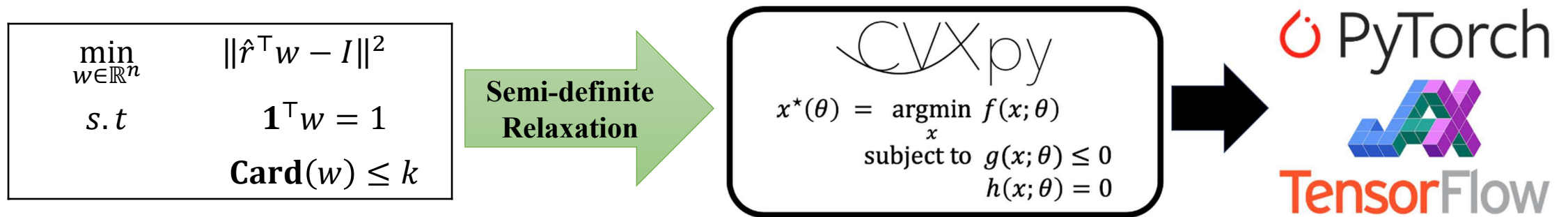
$\min_{W \in S_n^+}$	$\text{Tr}(\hat{R}W) - 2(\mathbf{1}^\top W \hat{R}C)$
	$\text{Tr}(I_N W) = 1$
	$\mathbf{1}^\top  W  \mathbf{1} \leq k \text{Tr}(W)$
	$W \in S_n^+$

## 2) Surrogate Loss Approach



## Semi-definite Relaxation Approach

- With semi-definite relaxation, we can transform the original problem to be convex
- Then, the CvxpyLayer can be used to obtain the gradients required for DFL



## Semi-definite reformulation

- We first introduce semi-definite matrix variable  $W = ww^T \in \mathbb{R}^{n \times n}$  and  $\hat{R} = \hat{r}\hat{r}^T \in \mathbb{R}^{n \times n}$

## Semi-definite reformulation

- We first introduce semi-definite matrix variable  $W = ww^\top \in \mathbb{R}^{n \times n}$  and  $\hat{R} = \hat{r}\hat{r}^\top \in \mathbb{R}^{n \times n}$
- The objective function can be reformulated as (**Tr**: Trace of matrix)

$$\|\hat{r}^\top w - I\|^2 = (\hat{r}^\top w)^2 - 2I(\hat{r}^\top w) + I^2 = \mathbf{Tr}(\hat{R}W) - 2I(\mathbf{1}^\top W \hat{r}) + I^2$$

## Semi-definite reformulation

- We first introduce semi-definite matrix variable  $W = ww^\top \in \mathbb{R}^{n \times n}$  and  $\hat{R} = \hat{r}\hat{r}^\top \in \mathbb{R}^{n \times n}$
- The objective function can be reformulated as (**Tr**: Trace of matrix)

$$\|\hat{r}^\top w - I\|^2 = (\hat{r}^\top w)^2 - 2I(\hat{r}^\top w) + I^2 = \text{Tr}(\hat{R}W) - 2I(\mathbf{1}^\top W \hat{r}) + I^2$$

- Since  $I^2$  is constant,



## Semi-definite reformulation

- Furthermore, the weight sum constraint and cardinality constraint are reformulated as

$\min_{w \in \mathbb{R}^n}$	$\ \hat{r}^\top w - I\ ^2$	$\rightarrow$	$\min_{W \in S_n^+}$	$\text{Tr}(\hat{R}W) - 2I(\mathbf{1}^\top W \hat{r})$
	$\mathbf{1}^\top w = 1$	$\rightarrow$		$\text{Tr}(I_N W) = 1$
	$\text{Card}(w) \leq k$	$\rightarrow$		$\text{Card}(W) \leq k^2$



## Semi-definite reformulation

- Finally,  $W = ww^T$  is equivalent (**Rank**( $W$ ) = 1 &  $W \in S_n^+$  (Positive semi-definite))

$\min_{w \in \mathbb{R}^n}$	$\ \hat{r}^T w - I\ ^2$	$\rightarrow$	$\min_{W \in S_n^+}$	$\text{Tr}(\hat{R}W) - 2I(\mathbf{1}^T W \hat{r})$
	$\mathbf{1}^T w = 1$	$\rightarrow$		$\text{Tr}(I_N W) = 1$
	$\text{Card}(w) \leq k$	$\rightarrow$		$\text{Card}(W) \leq k^2$
	$W = ww^T$	$\rightarrow$		$\text{Rank}(W) = 1$
				$W \in S_n^+$

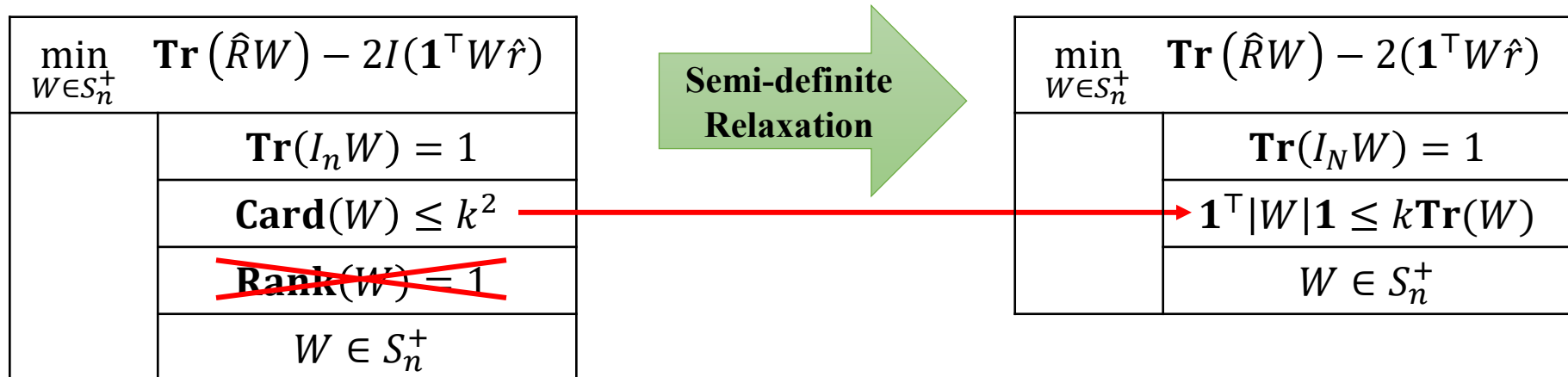
## Semi-definite relaxation

- Next, Let's relax the reformulated problem. We first drop the Rank constraint.

$\min_{W \in S_n^+}$	$\text{Tr}(\hat{R}W) - 2I(\mathbf{1}^\top W \hat{r})$
	$\text{Tr}(I_n W) = 1$
	$\text{Card}(W) \leq k^2$
	<del><math>\text{Rank}(W) = 1</math></del>
	$W \in S_n^+$

## Semi-definite relaxation (cont.)

- Next, by relaxing  $\mathbf{Card}(W) \leq k^2$  to  $\|w\|_1^2 \leq k\|w\|_2^2$  and transforming to  $\mathbf{1}^\top |W| \mathbf{1} \leq k\mathbf{Tr}(W)$ ,



## DPP-compliant

- When using CvxpyLayer, the formulation must satisfy DPP-compliant
  1. The optimization problem must be convex in variable  $W$  → **Already Satisfied**

## DPP-compliant

- When using CvxpyLayer, the formulation must satisfy DPP-compliant
  1. The optimization problem must be convex in variable  $W \rightarrow$  **Already Satisfied**
  2. The optimization problem must be linear in parameter  $\hat{R} \rightarrow$  **Use  $I = \hat{r}^\top C$  where  $C$  is market cap weight**

## DPP-compliant

- When using CvxpyLayer, the formulation must satisfy DPP-compliant
  - The optimization problem must be convex in variable  $W \rightarrow$  **Already Satisfied**
  - The optimization problem must be linear in parameter  $\hat{R} \rightarrow$  **Use  $I = \hat{r}^\top C$  where  $C$  is market cap weight**

$\min_{W \in S_n^+}$	$\text{Tr}(\hat{R}W) - 2I(\mathbf{1}^\top W \hat{r})$
	$\text{Tr}(I_N W) = 1$
	$\mathbf{1}^\top  W  \mathbf{1} \leq k \text{Tr}(W)$
	$W \in S_n^+$

$\min_{W \in S_n^+}$	$\text{Tr}(\hat{R}W) - 2(\mathbf{1}^\top W \hat{R} C)$
	$\text{Tr}(I_N W) = 1$
	$\mathbf{1}^\top  W  \mathbf{1} \leq k \text{Tr}(W)$
	$W \in S_n^+$

## DFL loss for Partial Index Tracking

- As seen in previous session, the DFL loss is expressed as

$$\begin{aligned}\mathcal{L}(\hat{R}, R) &:= \textit{Regret}(w^*(\hat{R}), R) \\ &= f(w^*(\hat{R}), R) - f(w^*(R), R)\end{aligned}$$

## DFL loss for Partial Index Tracking

- As seen in previous session, the DFL loss is expressed as

$$\begin{aligned}\mathcal{L}(\hat{R}, R) &:= \text{Regret}(w^*(\hat{R}), R) \\ &= \underline{f(w^*(\hat{R}), R)} - f(w^*(R), R)\end{aligned}$$

Realized decision quality with estimated  $\hat{R}$





## DFL loss for Partial Index Tracking

- As seen in previous session, the DFL loss is expressed as

$$\mathcal{L}(\hat{R}, R) := \text{Regret}(w^*(\hat{R}), R)$$

$$= \underline{f(w^*(\hat{R}), R)} - \underline{f(w^*(R), R)}$$

Realized decision quality with estimated  $\hat{R}$

Realized decision quality with ground truth  $R$

## DFL loss for Partial Index Tracking

- As seen in previous session, the DFL loss is expressed as

$$\mathcal{L}(\hat{R}, R) := \text{Regret}(w^*(\hat{R}), R)$$

$$= \underline{f(w^*(\hat{R}), R)} - \underline{f(w^*(R), R)}$$

Realized decision quality with estimated  $\hat{R}$

Realized decision quality with ground truth  $R$

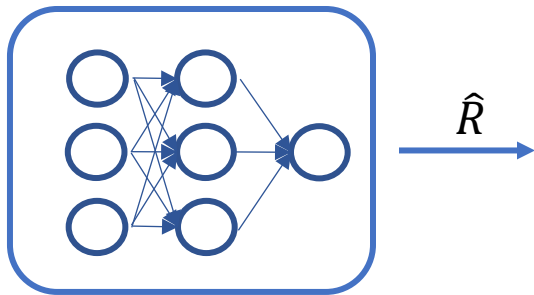
- Thus, in index tracking problem, the DFL loss is

$$\mathcal{L}(\hat{R}, R) = [\text{Tr}(RW^*(\hat{R})) - 2(\mathbf{1}^\top W^*(\hat{R})RC)] - [\text{Tr}(RW^*(R)) - 2(\mathbf{1}^\top W^*(R)RC)]$$

## DFL with semi-definite relaxation

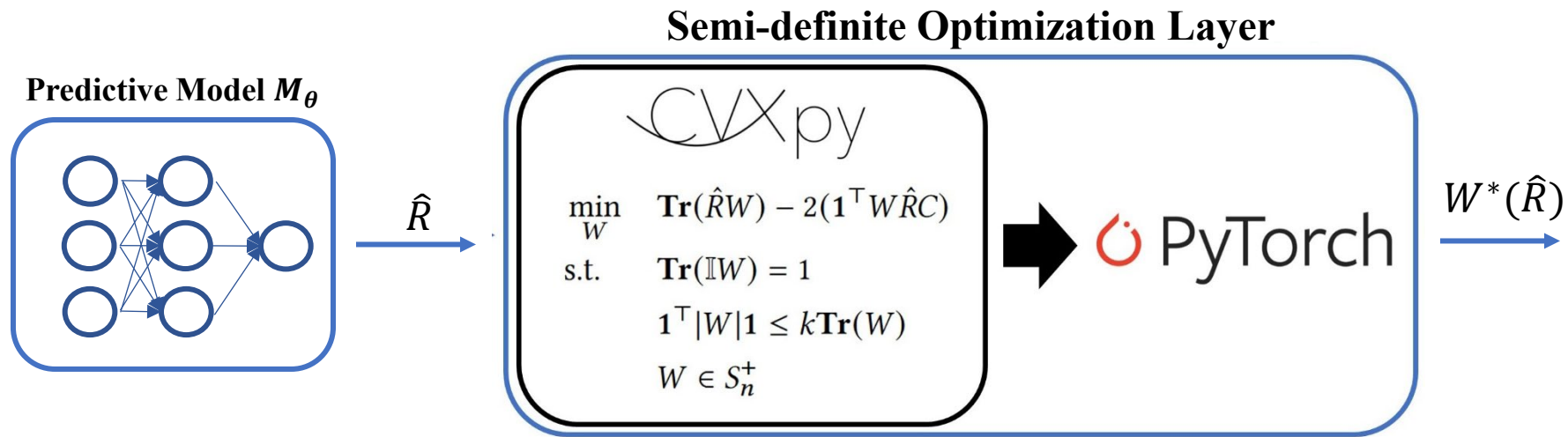
- The process of DFL in Partial Index Tracking

Predictive Model  $M_\theta$



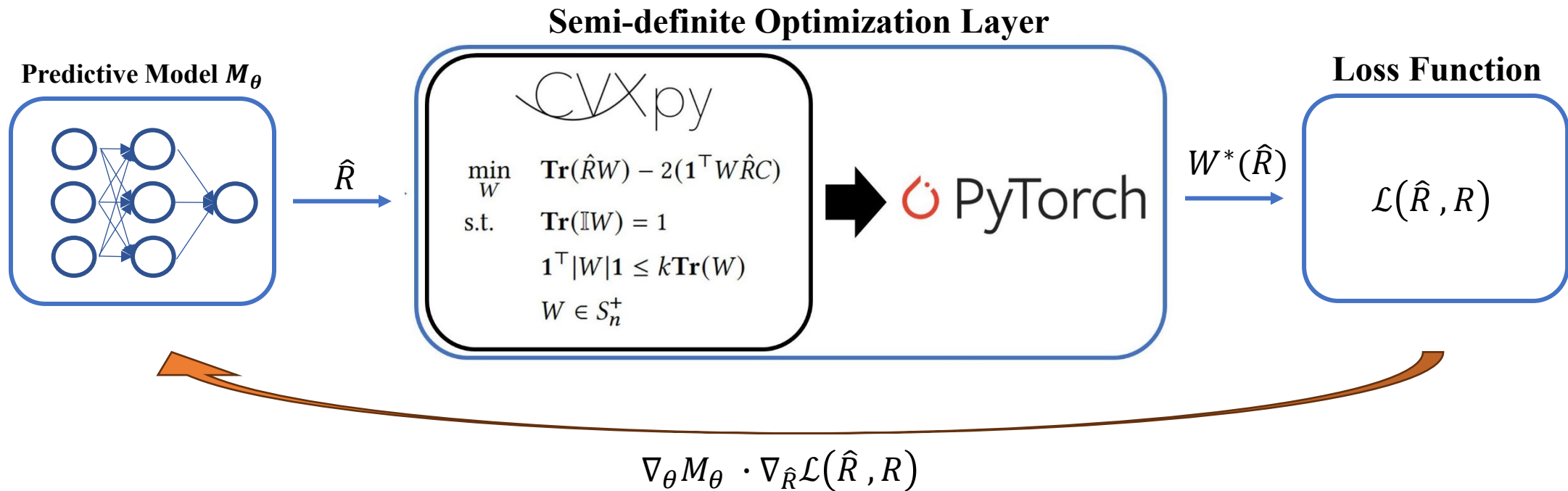
## DFL with semi-definite relaxation

- The process of DFL in Partial Index Tracking



## DFL with semi-definite relaxation

- The process of DFL in Partial Index Tracking



## Surrogate Approach

- The DFL loss function of basic formulation is

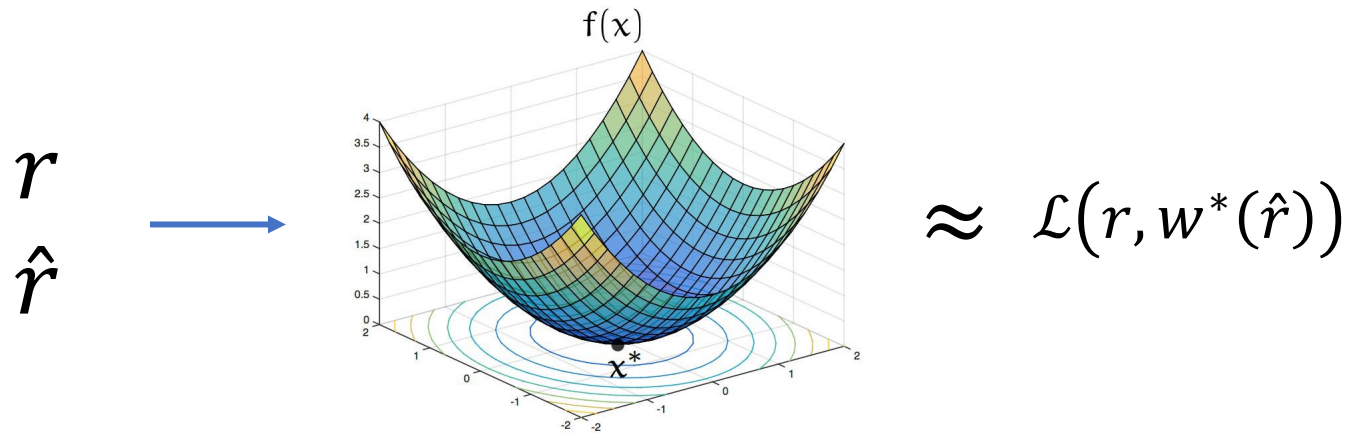
$$\mathcal{L}(\hat{r}, r) = \|r^\top w^*(\hat{r}) - I\|^2 - \|r^\top w^*(r) - I\|^2$$

## Surrogate Approach

- The DFL loss function of basic formulation is

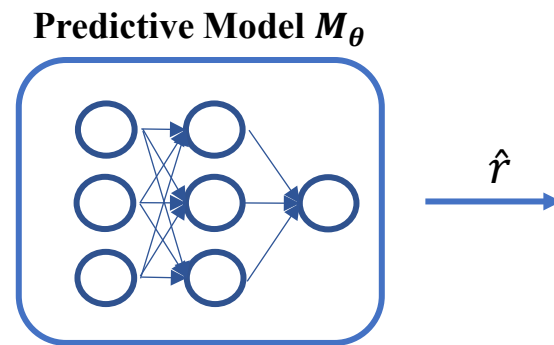
$$\mathcal{L}(\hat{r}, r) = \|r^\top w^*(\hat{r}) - I\|^2 - \|r^\top w^*(r) - I\|^2$$

- Surrogate approach aim to learn  $\mathcal{L}(\hat{r}, r)$  with another predictive model



## DFL with surrogate approach

- The process of DFL in Partial Index Tracking

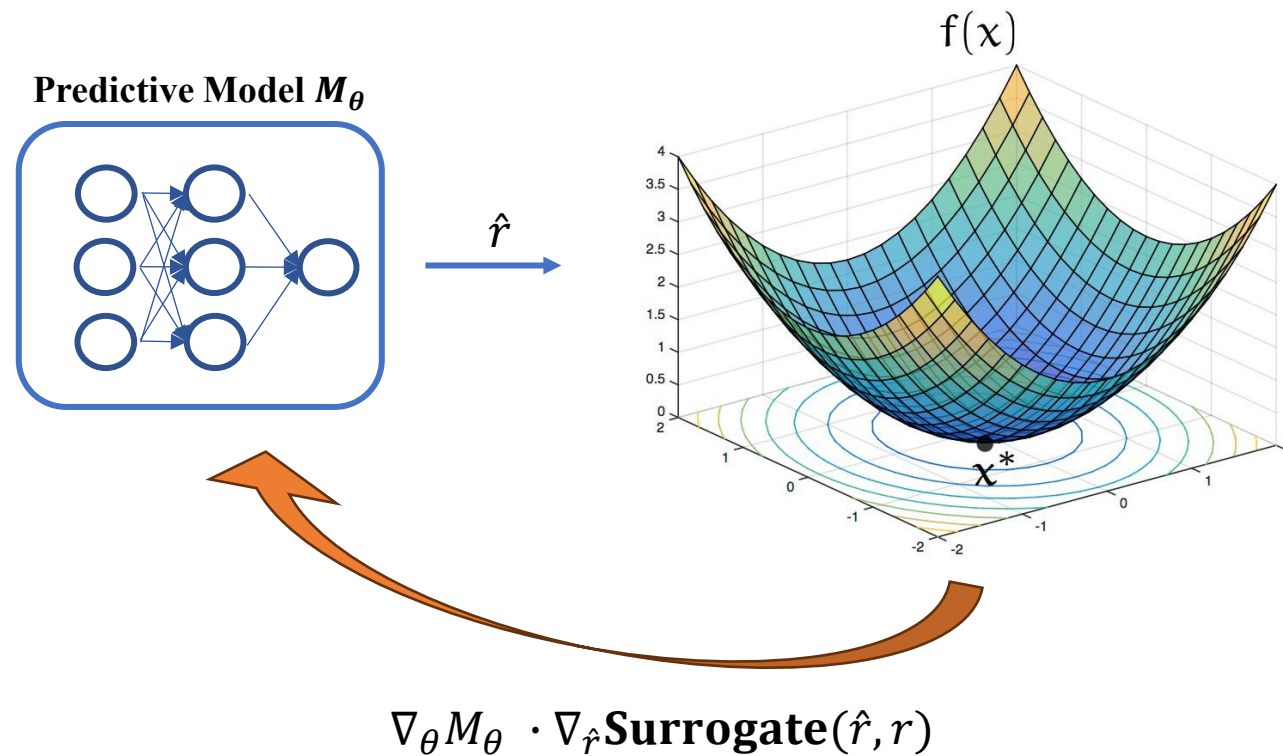




## DFL with surrogate approach

- The process of DFL in Partial Index Tracking

### Surrogate Loss Model



## Hands-on Exercise

- Partial Index Tracking problem with KOSPI index
  - For simplicity, with top 30 assets



## Data

- Closed price and Market cap weight of KOSPI30 (2015/01/02 ~ 2024/12/30)

## Data

- Closed price and Market cap weight of KOSPI30 (2015/01/02 ~ 2024/12/30)

## Features

Return before rebalancing day

Cumulative return over the past period

Mean and variance of past returns

Mean and variance of past prices

Current return

Current price

# Data

- Closed price and Market cap weight of KOSPI30 (2015/01/02 ~ 2024/12/30)

## Features

Return before rebalancing day

Cumulative return over the past period

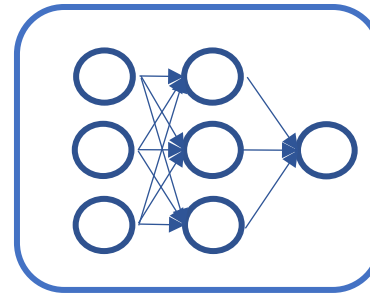
Mean and variance of past returns

Mean and variance of past prices

Current return

Current price

## Feed Forward Network



## Data

- Closed price and Market cap weight of KOSPI30 (2015/01/02 ~ 2024/12/30)

### Features

Return before rebalancing day

Cumulative return over the past period

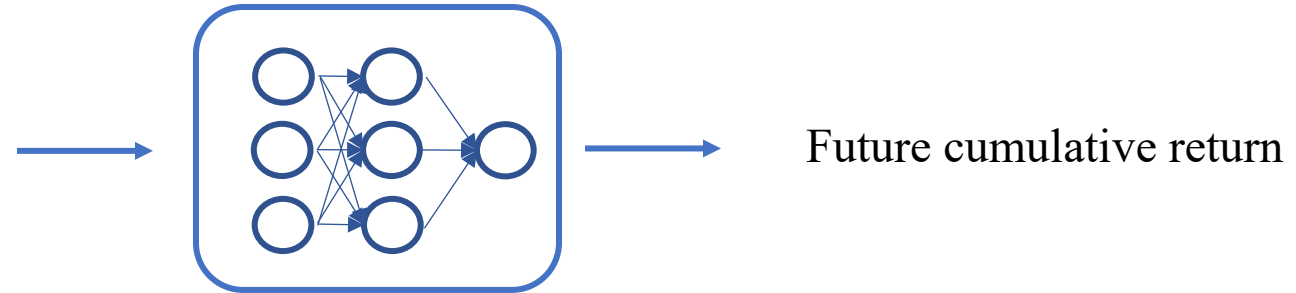
Mean and variance of past returns

Mean and variance of past prices

Current return

Current price

### Feed Forward Network



## **Hands-on Exercise**

- Let's implement partial index tracking directly with Colab!

### **Notebook & Data Link**





**Thank you for listening!**