The Benefits of Balance:

From Information Projections to Variance Reduction

Institute for Foundations of Data Science (IFDS) Seminar
April 18, 2025

Ronak Mehta

Team

Lang Liu
University of Washington

Ronak Mehta
University of Washington

Soumik Pal
University of Washington

Zaid Harchaoui
University of Washington

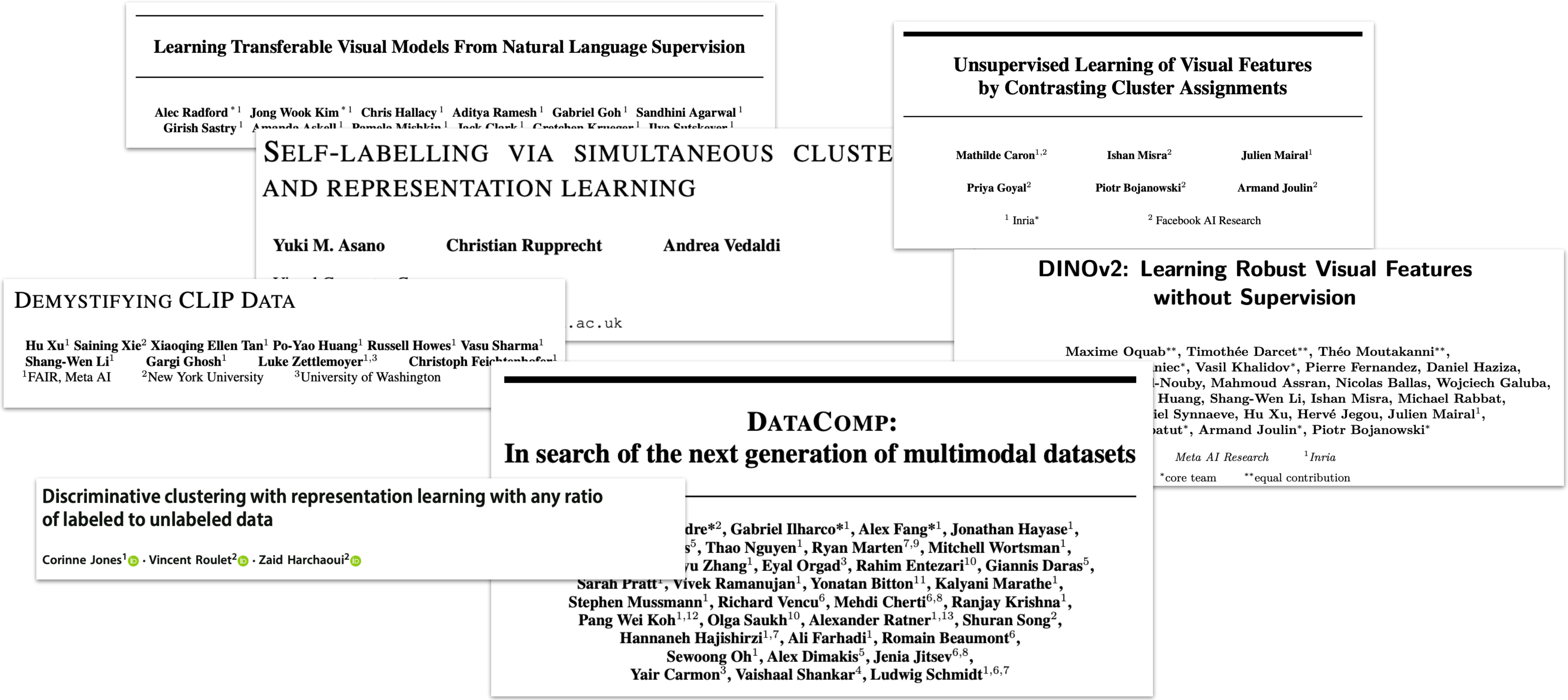
IFDS
Institute for Foundations of Data Science

NSF

IFML

W
UNIVERSITY *of* WASHINGTON

The Mystery of (Multimodal) Self-Supervised Learning



Learning Transferable Visual Models From Natural Language Supervision

Alec Radford *[1] Jong Wook Kim *[1] Chris Hallacy [1] Aditya Ramesh [1] Gabriel Goh [1] Sandhini Agarwal [1]
Girish Sastry [1] Amanda Askell [1] Pamela Mishkin [1] Jack Clark [1] Gretchen Krueger [1] Ilya Sutskever [1]

Unsupervised Learning of Visual Features by Contrasting Cluster Assignments

Mathilde Caron[1,2] Ishan Misra[2] Julien Mairal[1]
Priya Goyal[2] Piotr Bojanowski[2] Armand Joulin[2]
[1] Inria* [2] Facebook AI Research

SELF-LABELLING VIA SIMULTANEOUS CLUSTE
AND REPRESENTATION LEARNING

Yuki M. Asano Christian Rupprecht Andrea Vedaldi

.ac.uk

DINOv2: Learning Robust Visual Features without Supervision

Maxime Oquab**, Timothée Darcet**, Théo Moutakanni**,
niec*, Vasil Khalidov*, Pierre Fernandez, Daniel Haziza,
-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba,
Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat,
iel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal[1],
atut*, Armand Joulin*, Piotr Bojanowski*
Meta AI Research [1] *Inria*
*core team **equal contribution

DEMYSTIFYING CLIP DATA

Hu Xu[1] Saining Xie[2] Xiaoqing Ellen Tan[1] Po-Yao Huang[1] Russell Howes[1] Vasu Sharma[1]
Shang-Wen Li[1] Gargi Ghosh[1] Luke Zettlemoyer[1,3] Christoph Feichtenhofer[1]
[1]FAIR, Meta AI [2]New York University [3]University of Washington

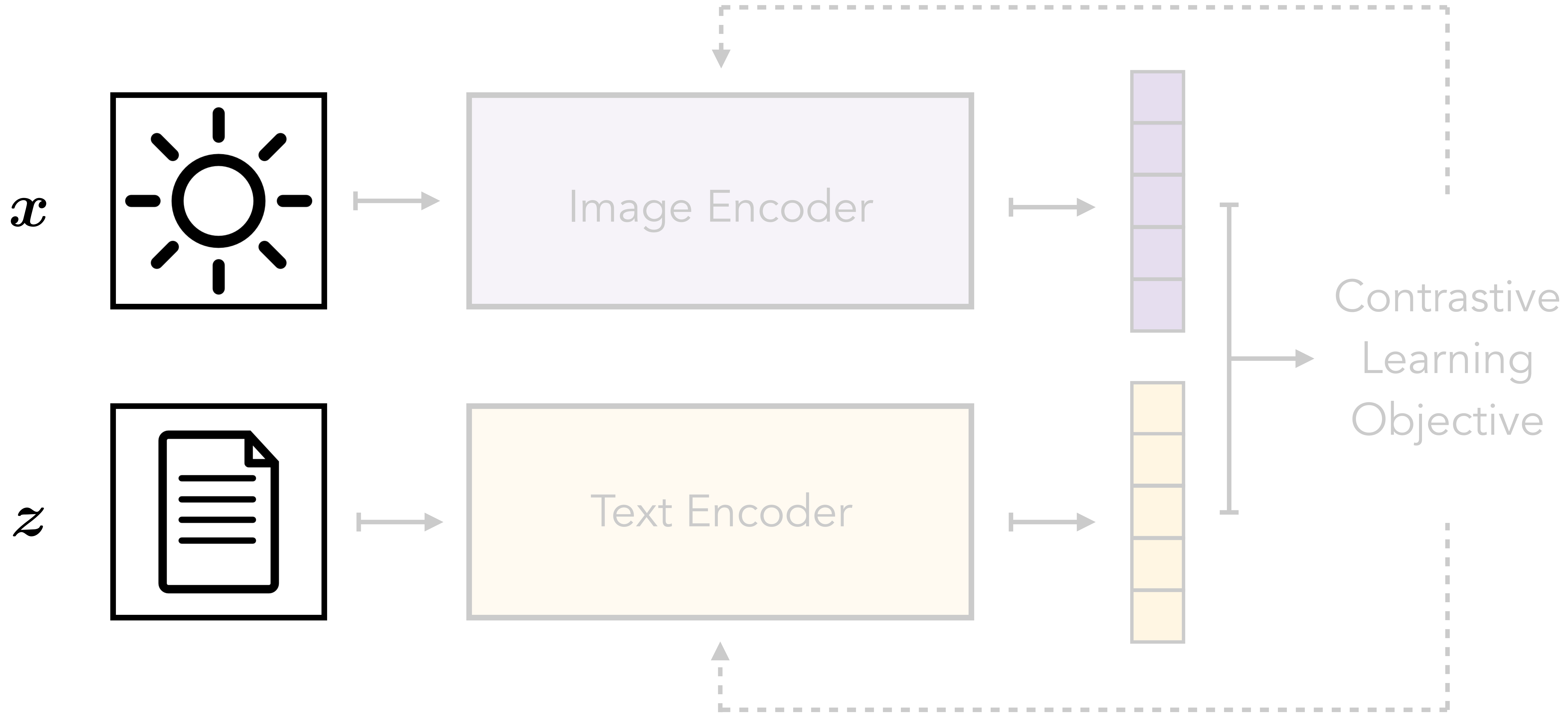
DATACOMP:
In search of the next generation of multimodal datasets

dre*[2], Gabriel Ilharco*[1], Alex Fang*[1], Jonathan Hayase[1],
s[5], Thao Nguyen[1], Ryan Marten[7,9], Mitchell Wortsman[1],
u Zhang[1], Eyal Orgad[3], Rahim Entezari[10], Giannis Daras[5],
Sarah Pratt[1], Vivek Ramanujan[1], Yonatan Bitton[11], Kalyani Marathe[1],
Stephen Mussmann[1], Richard Vencu[6], Mehdi Cherti[6,8], Ranjay Krishna[1],
Pang Wei Koh[1,12], Olga Saukh[10], Alexander Ratner[1,13], Shuran Song[2],
Hannaneh Hajishirzi[1,7], Ali Farhadi[1], Romain Beaumont[6],
Sewoong Oh[1], Alex Dimakis[5], Jenia Jitsev[6,8],
Yair Carmon[3], Vaishaal Shankar[4], Ludwig Schmidt[1,6,7]

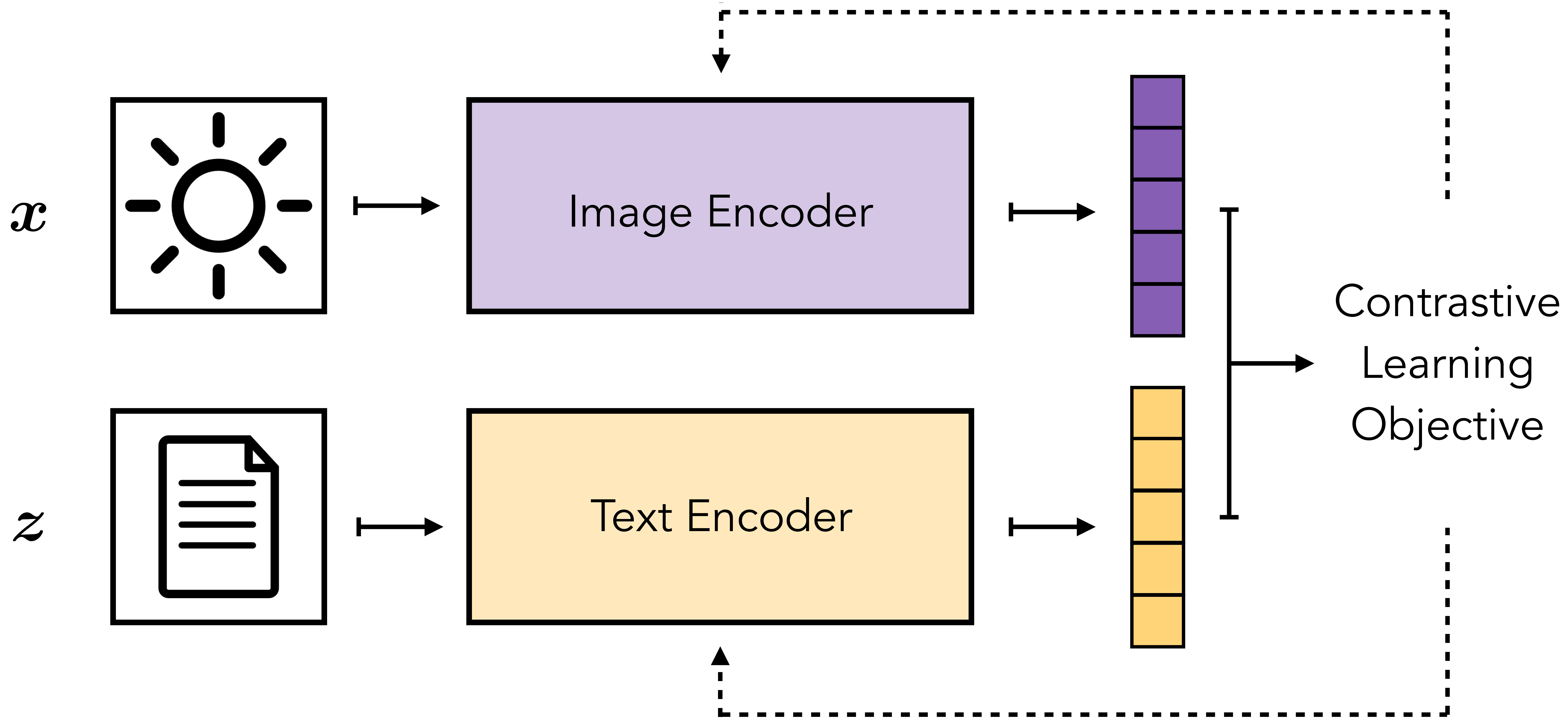
Discriminative clustering with representation learning with any ratio of labeled to unlabeled data

Corinne Jones[1] · Vincent Roulet[2] · Zaid Harchaoui[2]

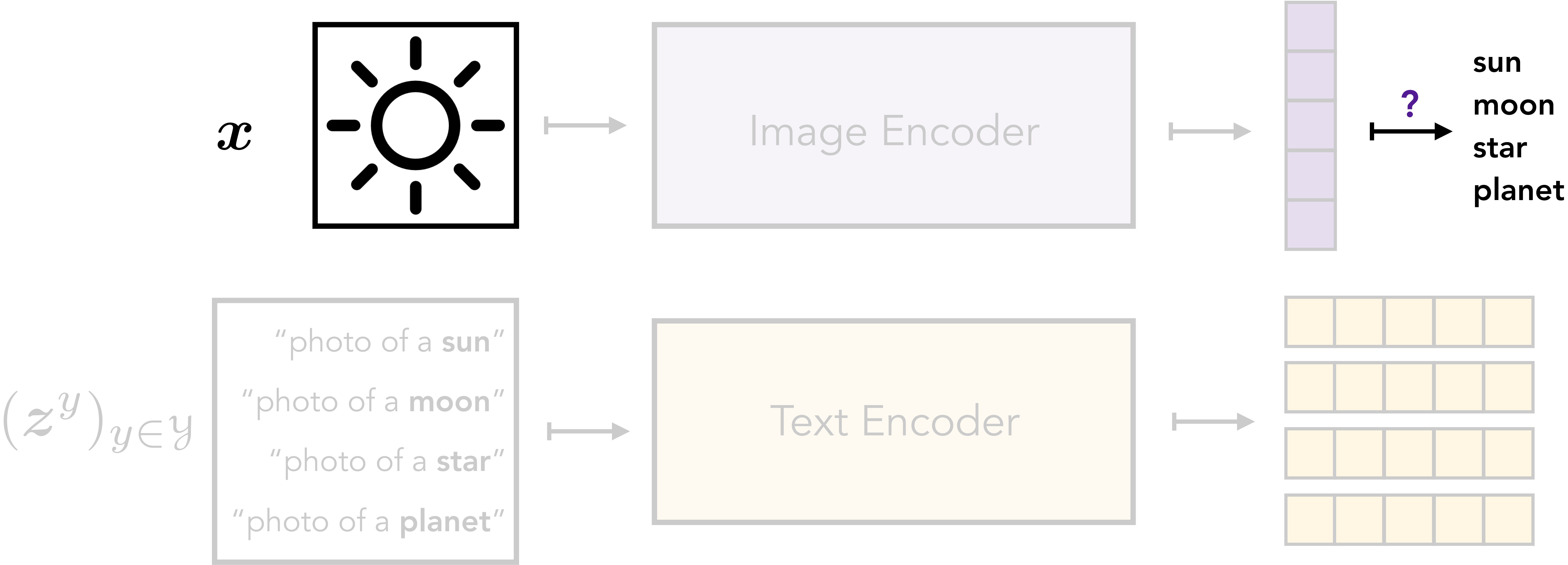
Pre-Training: Self-Supervised Learning



Pre-Training: Self-Supervised Learning

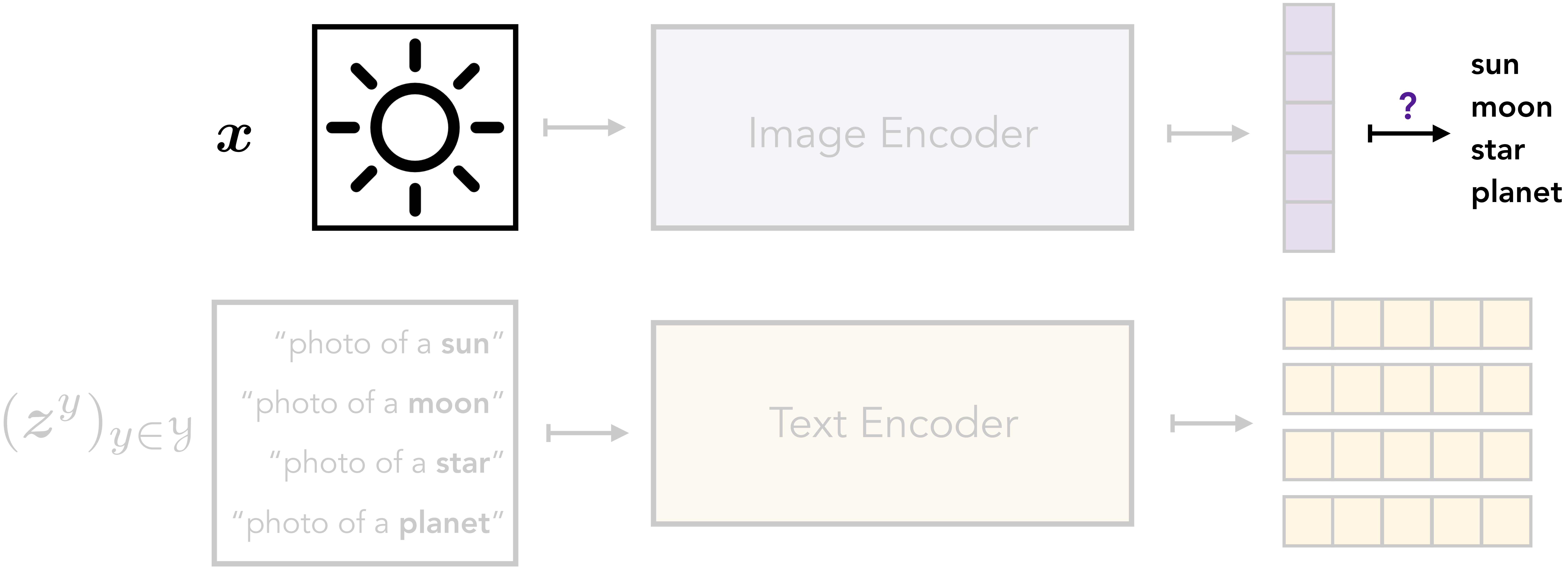


Inference: Prompting (Zero-Shot)

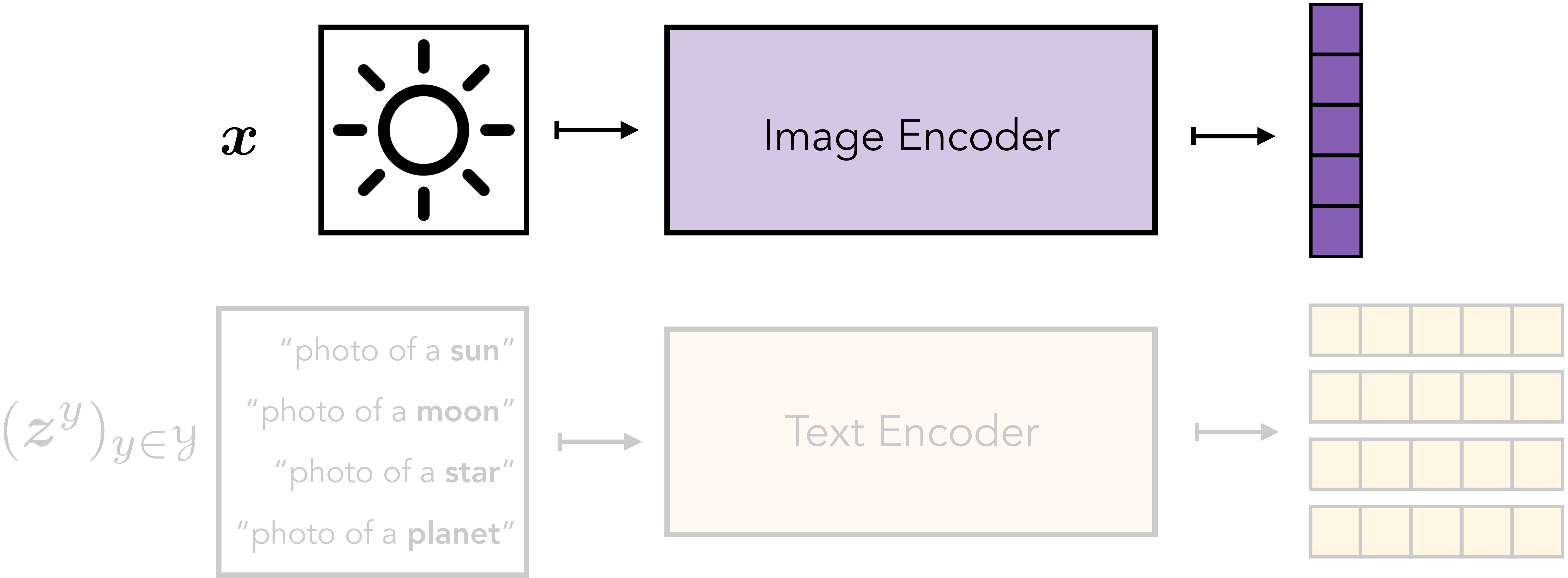


Inference: Prompting (Zero-Shot)

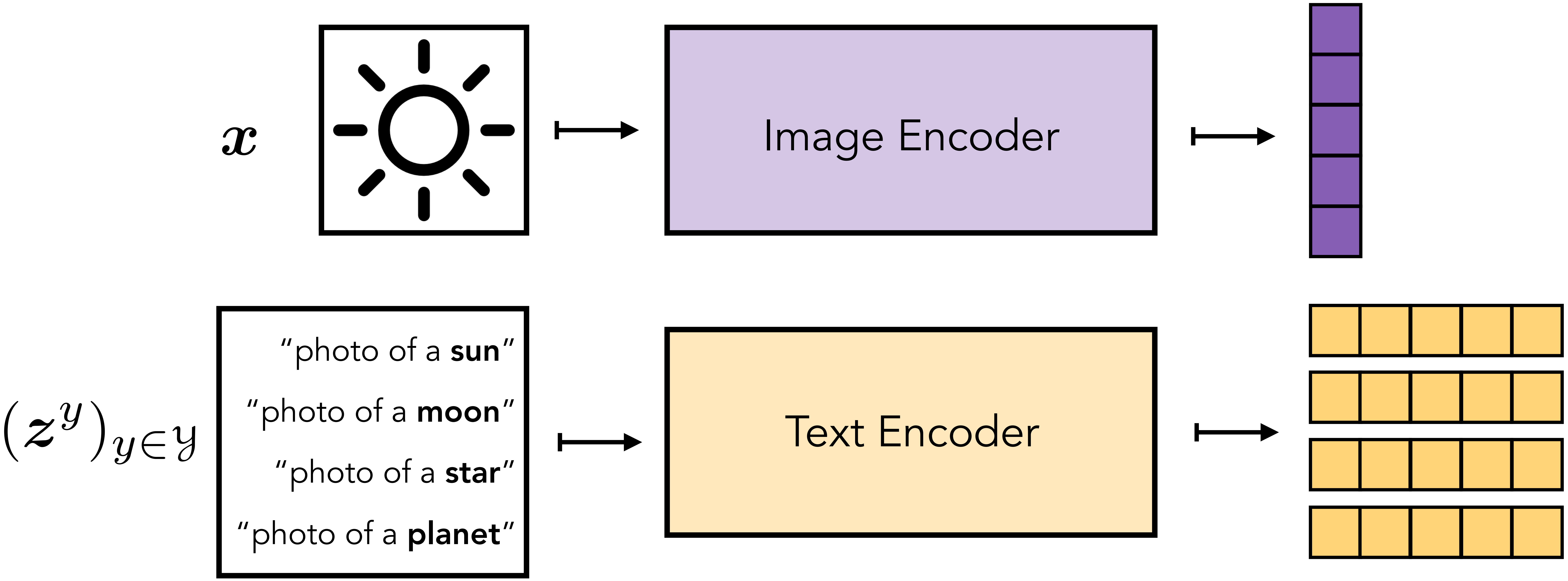
No directly labeled training data supplied to user.



Inference: Prompting (Zero-Shot)



Inference: Prompting (Zero-Shot)



$\boldsymbol{x}$

Image Encoder

$(\boldsymbol{z}^y)_{y\in\mathcal{Y}}$

"photo of a **sun**"

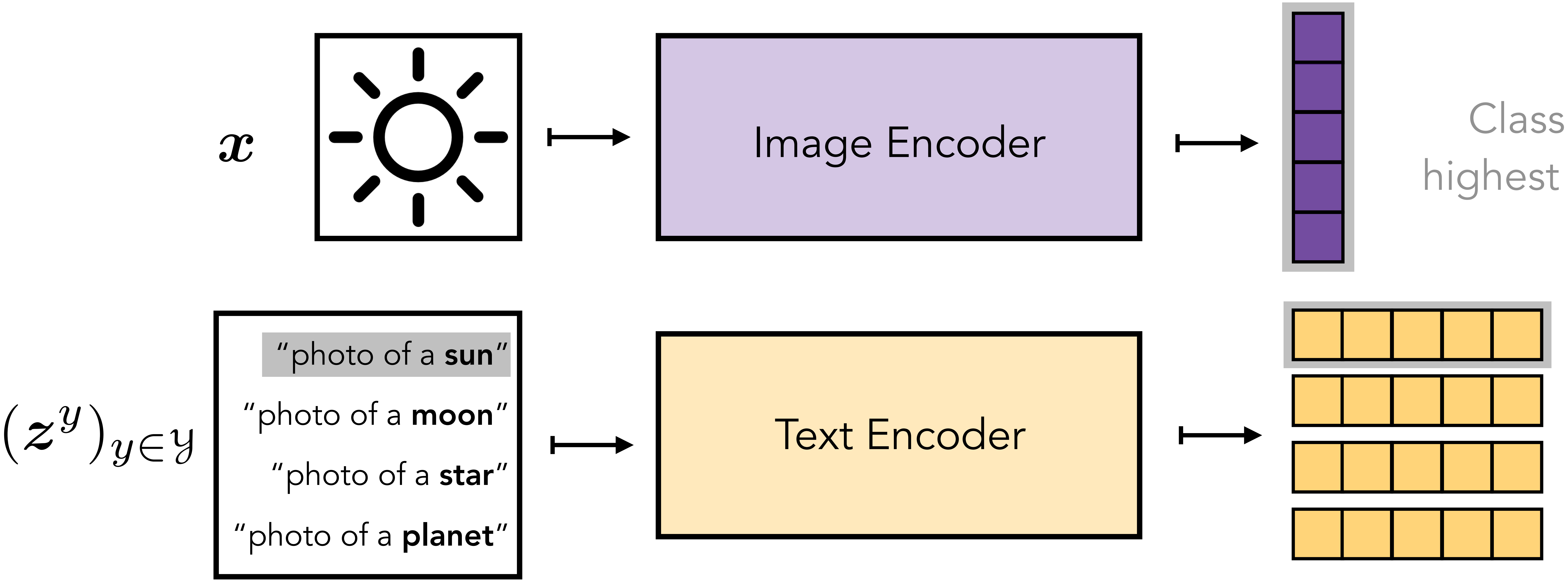
"photo of a **moon**"

"photo of a **star**"

"photo of a **planet**"

Text Encoder

Inference: Prompting (Zero-Shot)



$\boldsymbol{x}$

Image Encoder

$(\boldsymbol{z}^y)_{y\in\mathcal{Y}}$

"photo of a **sun**"

"photo of a **moon**"

"photo of a **star**"

"photo of a **planet**"

Text Encoder

Classified label = highest inner product.

Three Ingredients of Success

Pre-Training Data

Self-Supervised Learning Objective

Prompting/ Pseudo-Captioning

Three Ingredients of Success

Pre-Training Data

Self-Supervised Learning Objective

Prompting/ Pseudo-Captioning

What is the effect of common multimodal data curation methods on pre-training/downstream performance?

Three Ingredients of Success

Pre-Training Data

Self-Supervised Learning Objective

Prompting/ Pseudo-Captioning

What is the effect of common multimodal data curation methods on pre-training/downstream performance?

How do we interpret the CLIP objective (large batch limit, etc.) and improve it?

Three Ingredients of Success

Pre-Training Data

Self-Supervised Learning Objective

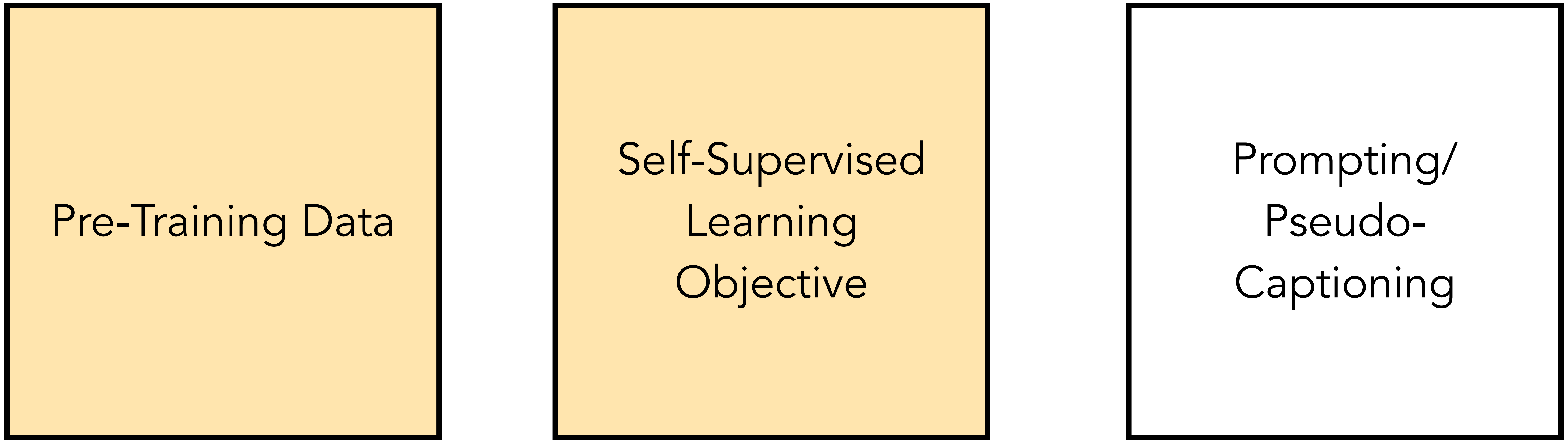
Prompting/ Pseudo-Captioning

What is the effect of common multimodal data curation methods on pre-training/downstream performance?

How do we interpret the CLIP objective (large batch limit, etc.) and improve it?

When can prompt-based zero-shot prediction match the performance of supervised learning?

Three Ingredients of Success



Pre-Training Data

Self-Supervised Learning Objective

Prompting/ Pseudo-Captioning

What is the effect of common multimodal data curation methods on pre-training/downstream performance?

How do we interpret the CLIP objective (large batch limit, etc.) and improve it?

When can prompt-based zero-shot prediction match the performance of supervised learning?

We will show that the key to both questions will be a connection to a decades-old statistics problem.

$$(X_1, Z_1), \ldots, (X_n, Z_n) \sim P$$

Marginals Distributions (P_X, P_Z)

We will show that the key to both questions will be a connection to a decades-old statistics problem.

$$(X_1, Z_1), \ldots, (X_n, Z_n) \sim P$$

Marginals Distributions (P_X, P_Z)

Using the **known** marginals, can we better estimate the **unknown** joint distribution?

How do we incorporate the marginal information and what do we gain?

We will show that the key to both questions will be a connection to a decades-old statistics problem.

$$(X_1, Z_1), \ldots, (X_n, Z_n) \sim P$$

Marginals Distributions (P_X, P_Z)

Using the **known** marginals, can we better estimate the **unknown** joint distribution?

How do we incorporate the marginal information and what do we gain?

Test Function $h : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$

Estimand $P(h) := \mathbb{E}_P\left[h(X, Z)\right]$

Empirical Measure $P_n := \frac{1}{n}\sum_{i=1}^{n} \delta_{(X_i, Z_i)}$

We will show that the key to both questions will be a connection to a decades-old statistics problem.

$$(X_1, Z_1), \ldots, (X_n, Z_n) \sim P$$

Marginals Distributions (P_X, P_Z)

Using the **known** marginals, can we better estimate the **unknown** joint distribution?

How do we incorporate the marginal information and what do we gain?

Test Function $h : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$

Estimand $P(h) := \mathbb{E}_P\left[h(X, Z)\right]$

Empirical Measure $P_n := \frac{1}{n} \sum_{i=1}^{n} \delta_{(X_i, Z_i)}$

Can we improve upon the standard estimator

$$P_n(h) = \frac{1}{n} \sum_{i=1}^{n} h(X_i, Z_i)$$

in terms of mean squared error?

Marginals are incorporated by **data balancing.**

(Sinkhorn Iterations, Iterative Proportional Fitting, Raking Ratio Estimation)

Marginals are incorporated by **data balancing.**

(Sinkhorn Iterations, Iterative Proportional Fitting, Raking Ratio Estimation)

$$P_n^{(0)} = P_n$$

$$P_n^{(k)} = \begin{cases} \arg\min_{Q : Q_X = P_X} \mathrm{KL}(Q \| P_n^{(k-1)}) & k \text{ odd} \\ \arg\min_{Q : Q_Y = P_Y} \mathrm{KL}(Q \| P_n^{(k-1)}) & k \text{ even} \end{cases}$$

Marginals are incorporated by **data balancing.**

(Sinkhorn Iterations, Iterative Proportional Fitting, Raking Ratio Estimation)

$$P_n^{(0)} = P_n$$

$$P_n^{(k)} = \begin{cases} \arg\min_{Q : Q_X = P_X} \mathrm{KL}(Q \| P_n^{(k-1)}) & k \text{ odd} \\ \arg\min_{Q : Q_Y = P_Y} \mathrm{KL}(Q \| P_n^{(k-1)}) & k \text{ even} \end{cases}$$

Odd Iterations

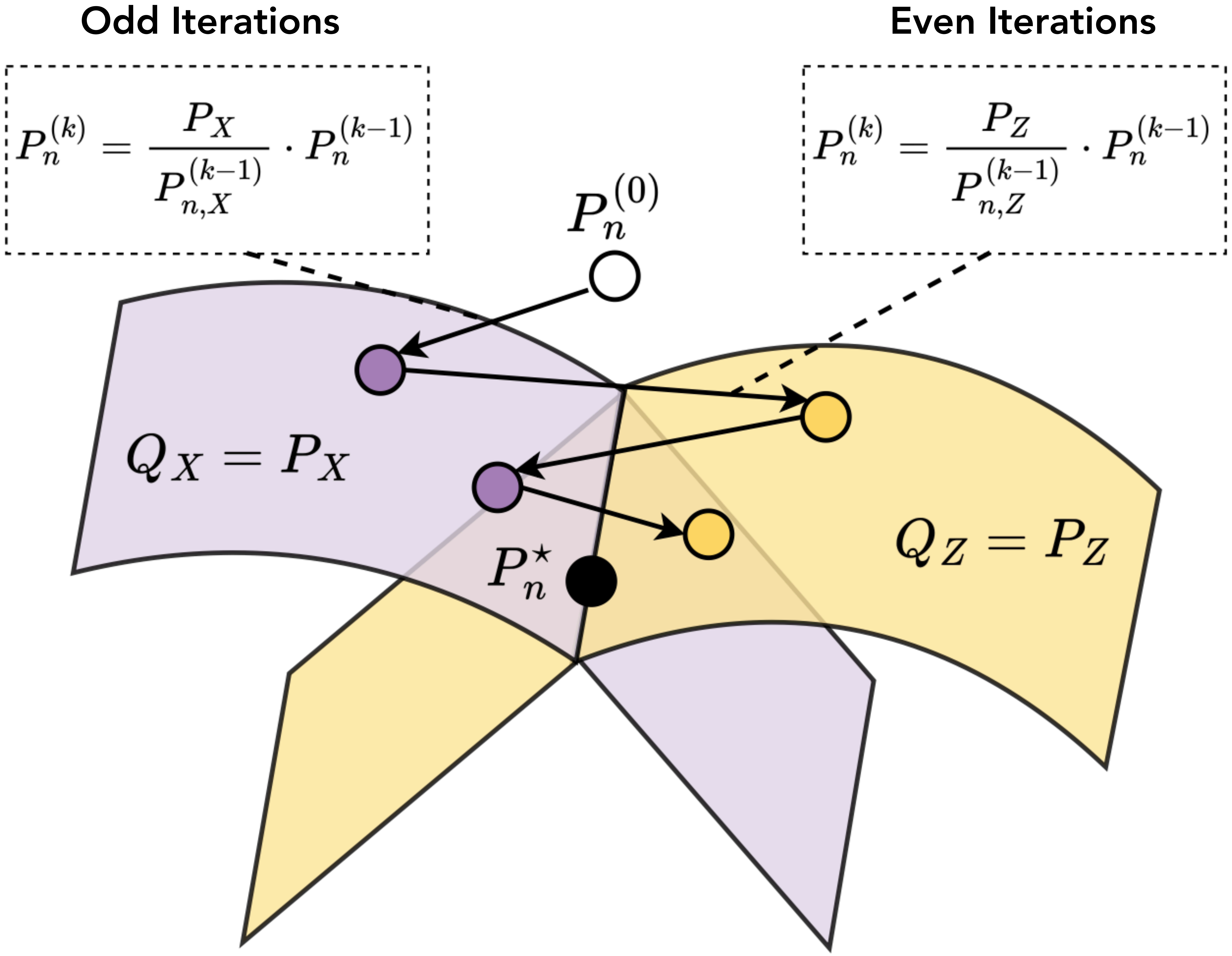
$$P_n^{(k-1)} \mapsto \frac{P_X}{P_{n,X}^{(k-1)}} \otimes P_n^{(k-1)}$$

Even Iterations

$$P_n^{(k-1)} \mapsto \frac{P_Z}{P_{n,Z}^{(k-1)}} \otimes P_n^{(k-1)}$$

Marginals are incorporated by **data balancing.**

(Sinkhorn Iterations, Iterative Proportional Fitting, Raking Ratio Estimation)


Odd Iterations
$P_n^{(k)} = \frac{P_X}{P_{n,X}^{(k-1)}} \cdot P_n^{(k-1)}$
Even Iterations
$P_n^{(k)} = \frac{P_Z}{P_{n,Z}^{(k-1)}} \cdot P_n^{(k-1)}$
$P_n^{(0)}$
$Q_X = P_X$
$P_n^\star$
$Q_Z = P_Z$

Marginals are incorporated by **data balancing.**

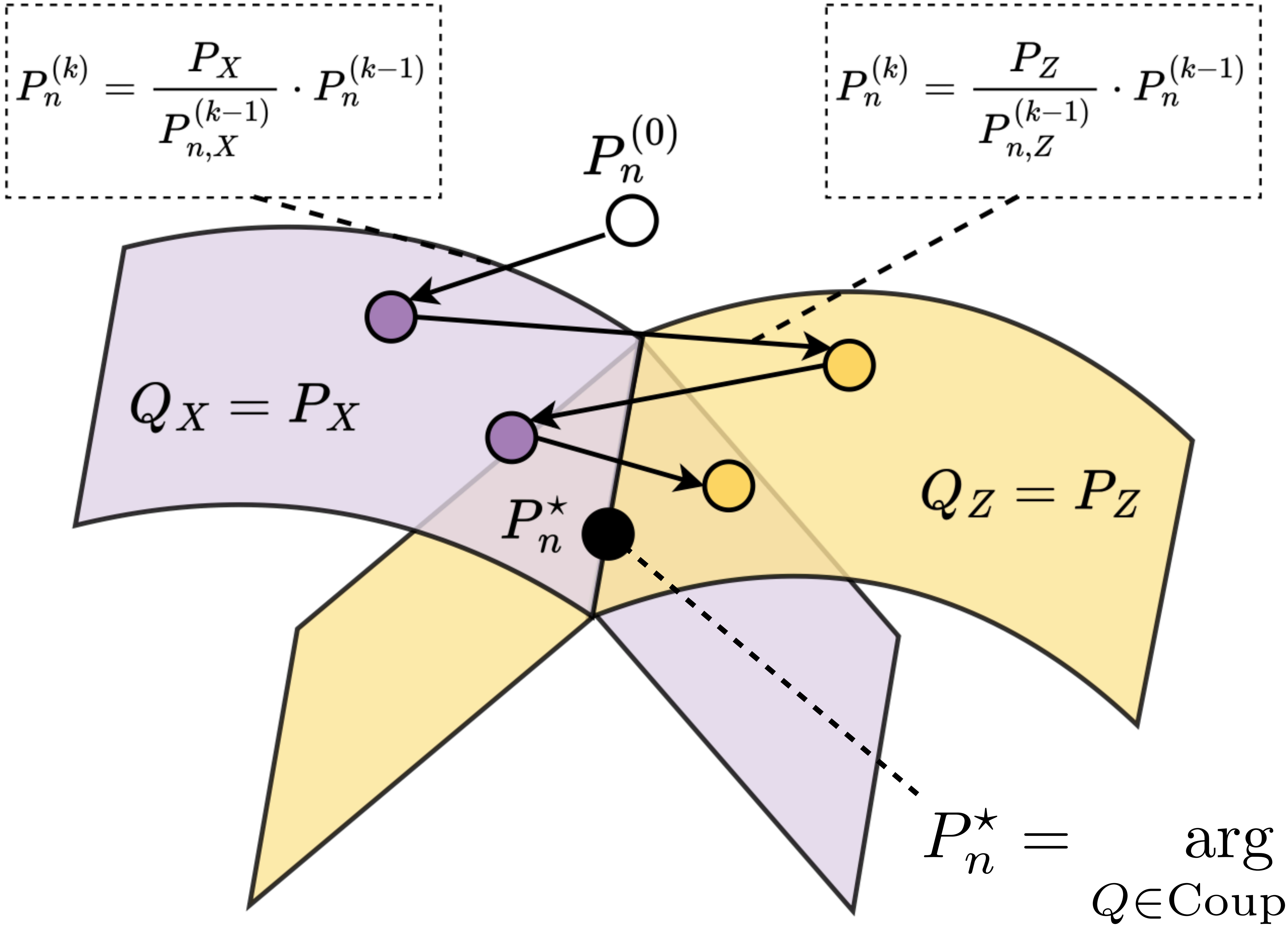
(Sinkhorn Iterations, Iterative Proportional Fitting, Raking Ratio Estimation)

Odd Iterations

$$P_n^{(k)} = \frac{P_X}{P_{n,X}^{(k-1)}} \cdot P_n^{(k-1)}$$

Even Iterations

$$P_n^{(k)} = \frac{P_Z}{P_{n,Z}^{(k-1)}} \cdot P_n^{(k-1)}$$

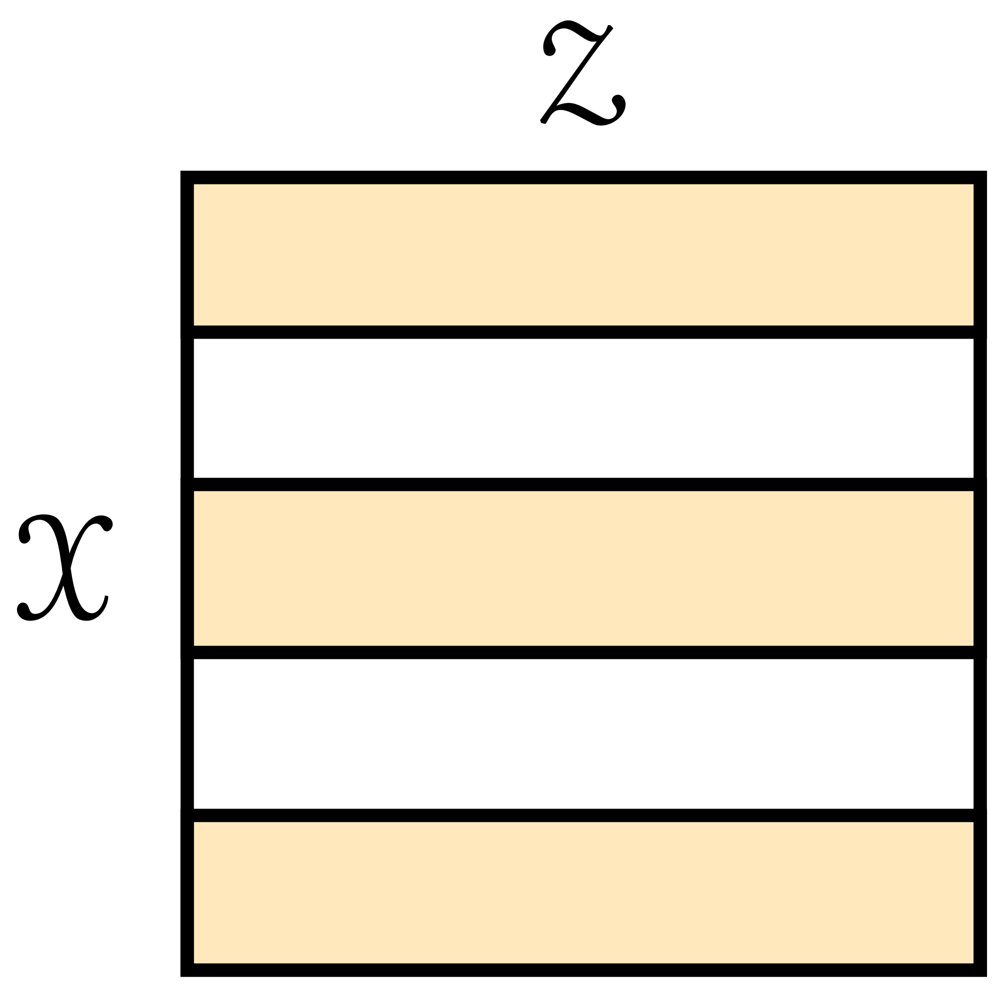


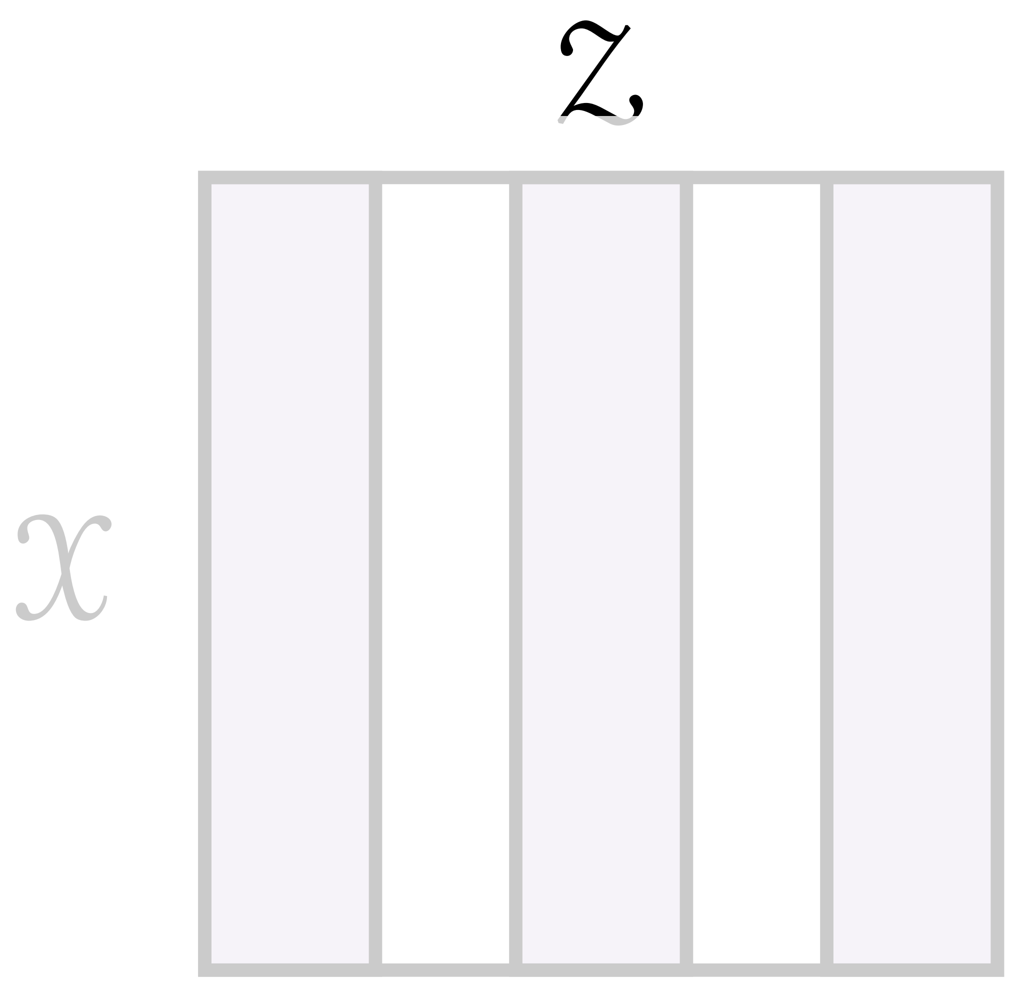
$$P_n^\star = \underset{Q \in \mathrm{Coup}(P_X, P_Z)}{\arg\min} \; \mathrm{KL}(Q \| P_n)$$

Marginals are incorporated by **data balancing.**

(Sinkhorn Iterations, Iterative Proportional Fitting, Raking Ratio Estimation)

$P_n^{(k-1)}$





Odd Iterations

$$P_n^{(k-1)} \mapsto \frac{P_X}{P_{n,X}^{(k-1)}} \otimes P_n^{(k-1)}$$

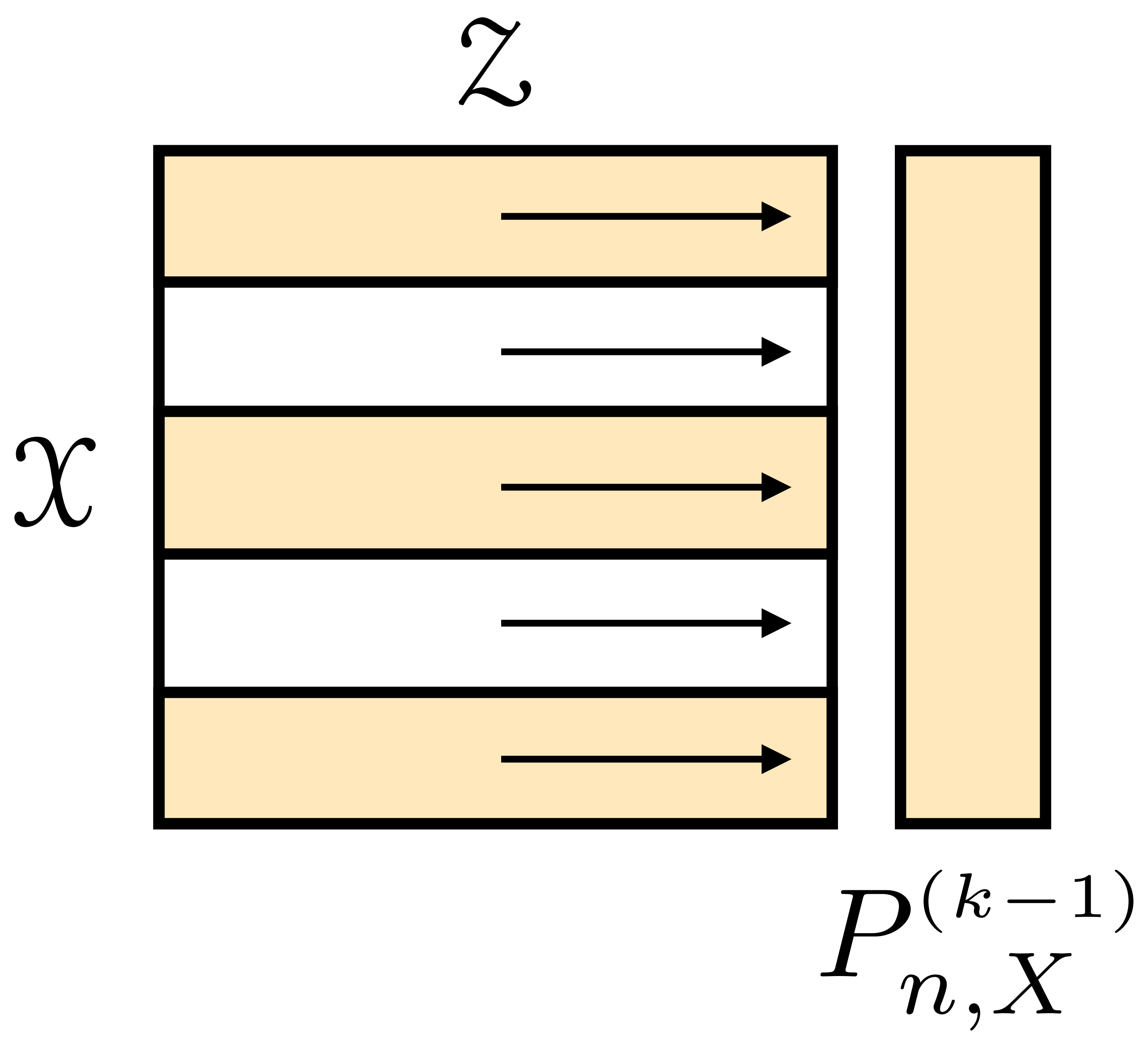
Even Iterations

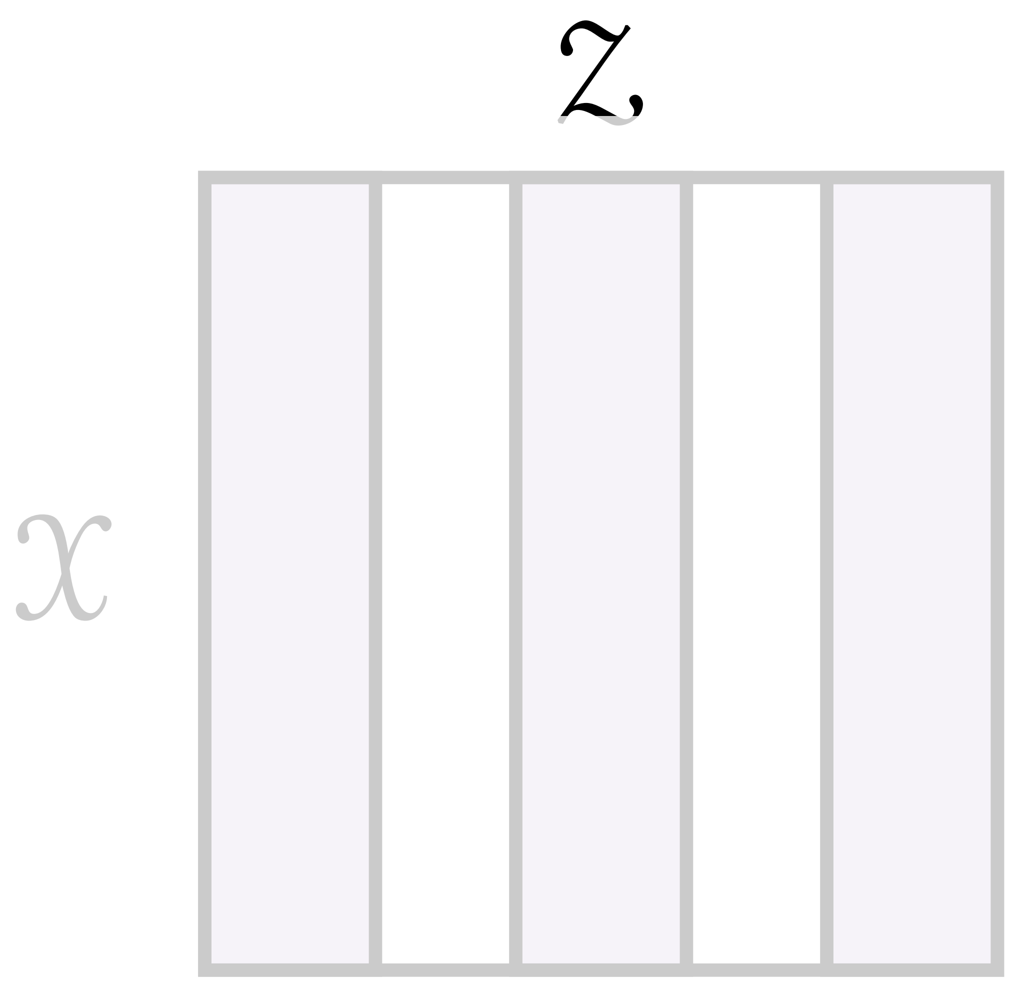
$$P_n^{(k-1)} \mapsto \frac{P_Z}{P_{n,Z}^{(k-1)}} \otimes P_n^{(k-1)}$$

Marginals are incorporated by **data balancing.**

(Sinkhorn Iterations, Iterative Proportional Fitting, Raking Ratio Estimation)

$P_n^{(k-1)}$





Odd Iterations

$$P_n^{(k-1)} \mapsto \frac{P_X}{P_{n,X}^{(k-1)}} \otimes P_n^{(k-1)}$$

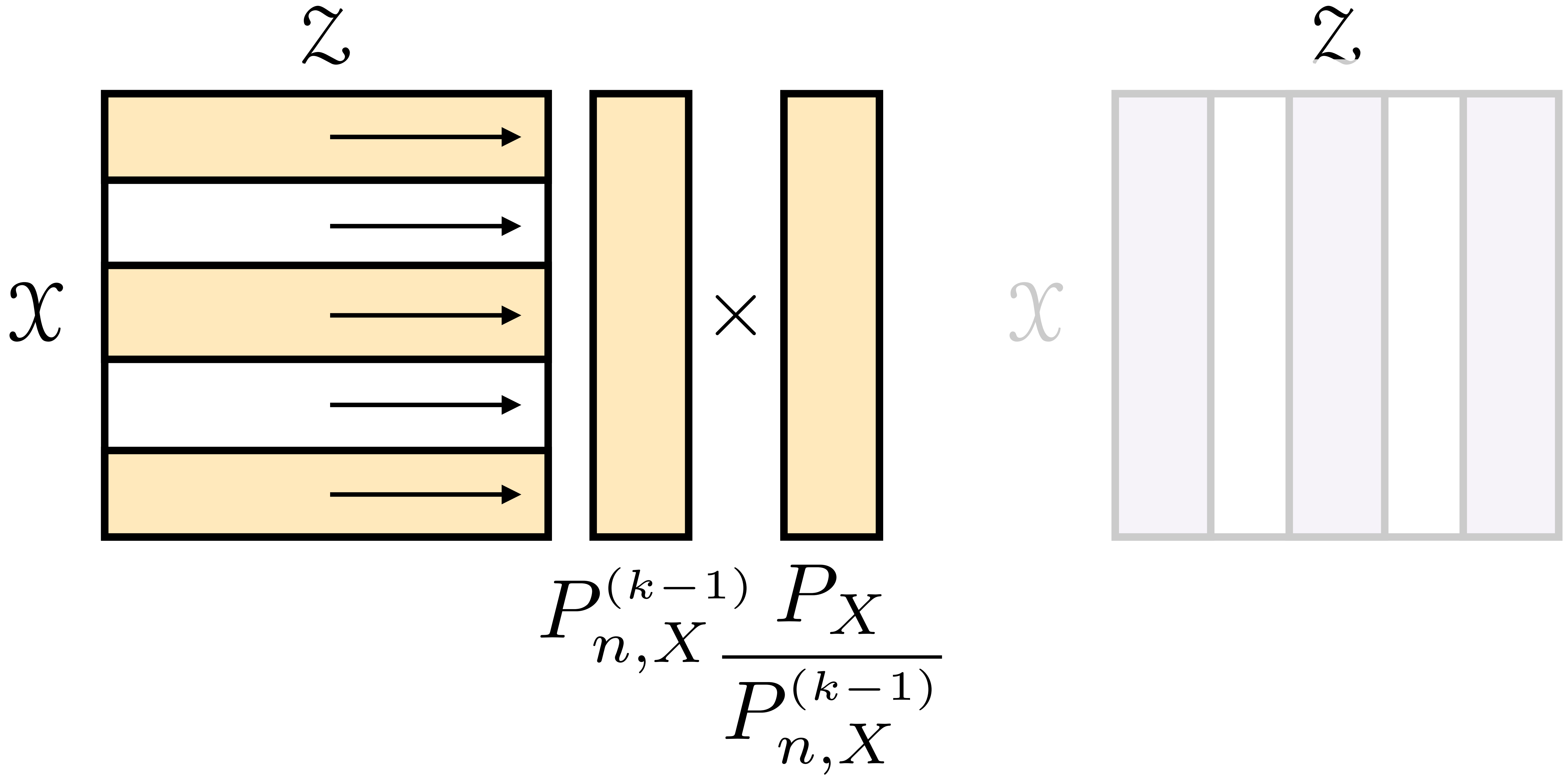
Even Iterations

$$P_n^{(k-1)} \mapsto \frac{P_Z}{P_{n,Z}^{(k-1)}} \otimes P_n^{(k-1)}$$

Marginals are incorporated by **data balancing.**

(Sinkhorn Iterations, Iterative Proportional Fitting, Raking Ratio Estimation)

$P_n^{(k-1)}$



Odd Iterations

$$P_n^{(k-1)} \mapsto \frac{P_X}{P_{n,X}^{(k-1)}} \otimes P_n^{(k-1)}$$

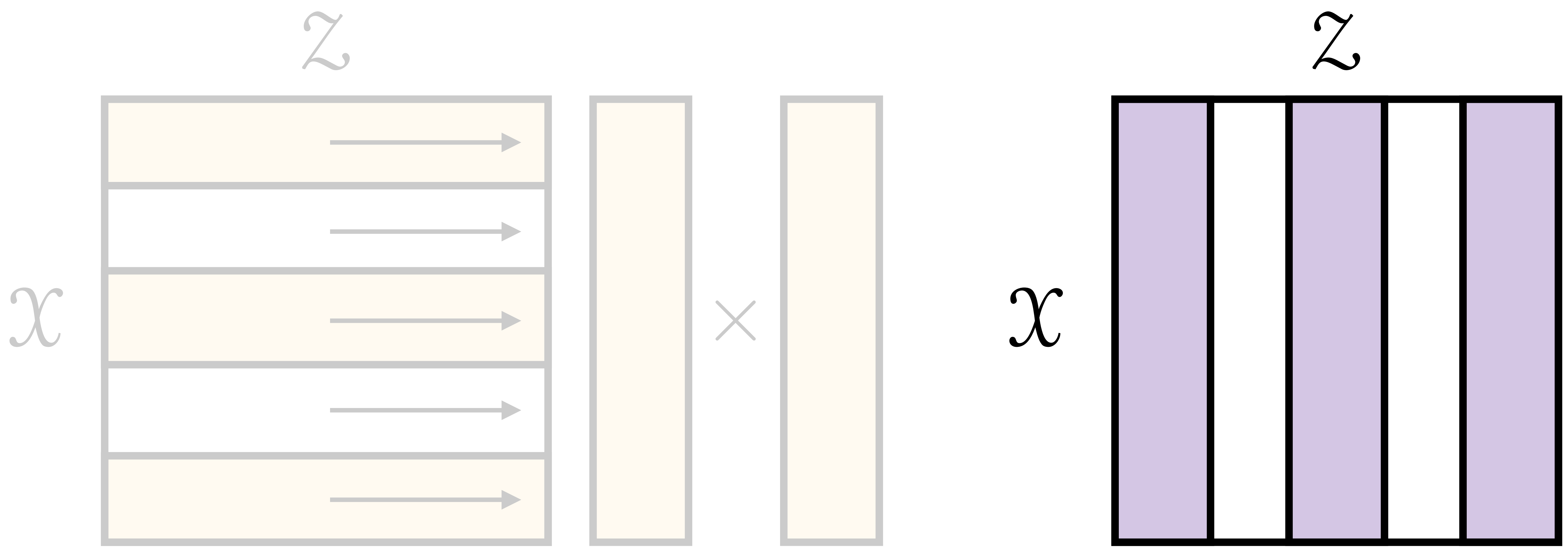
Even Iterations

$$P_n^{(k-1)} \mapsto \frac{P_Z}{P_{n,Z}^{(k-1)}} \otimes P_n^{(k-1)}$$

Marginals are incorporated by **data balancing.**

(Sinkhorn Iterations, Iterative Proportional Fitting, Raking Ratio Estimation)

$P_n^{(k-1)}$



$P_{n,X}^{(k-1)}$ $\frac{P_X}{P_{n,X}^{(k-1)}}$

Odd Iterations

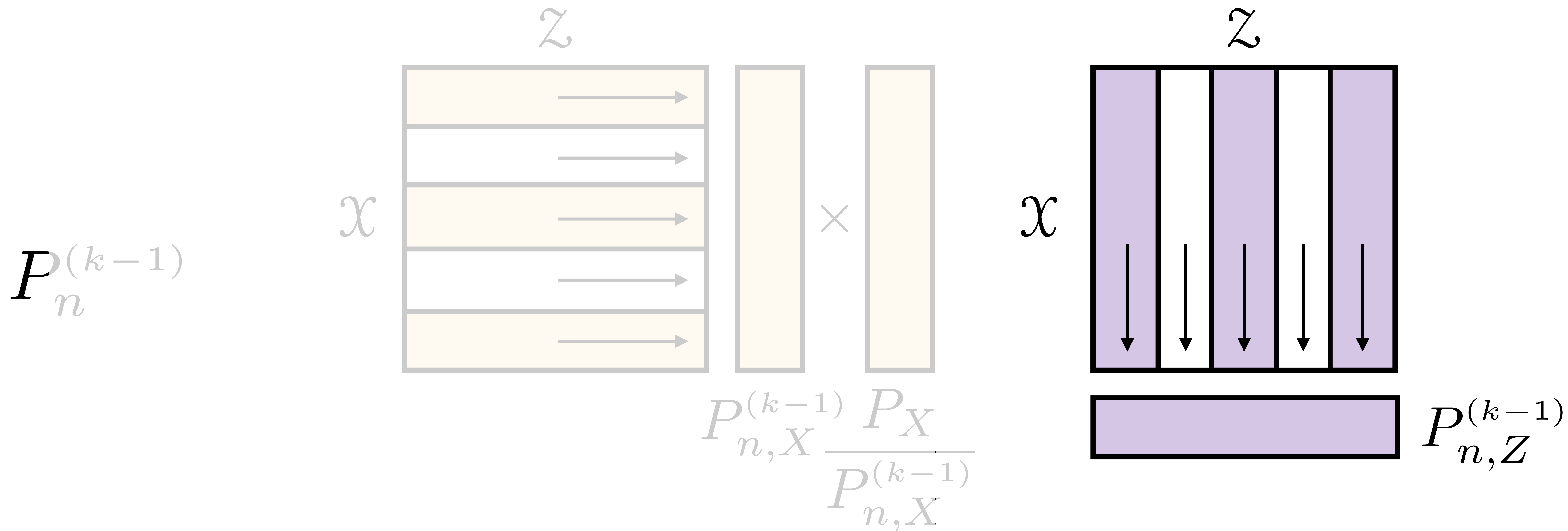
$$P_n^{(k-1)} \mapsto \frac{P_X}{P_{n,X}^{(k-1)}} \otimes P_n^{(k-1)}$$

Even Iterations

$$P_n^{(k-1)} \mapsto \frac{P_Z}{P_{n,Z}^{(k-1)}} \otimes P_n^{(k-1)}$$

Marginals are incorporated by **data balancing.**

(Sinkhorn Iterations, Iterative Proportional Fitting, Raking Ratio Estimation)



Odd Iterations

$$P_n^{(k-1)} \mapsto \frac{P_X}{P_{n,X}^{(k-1)}} \otimes P_n^{(k-1)}$$

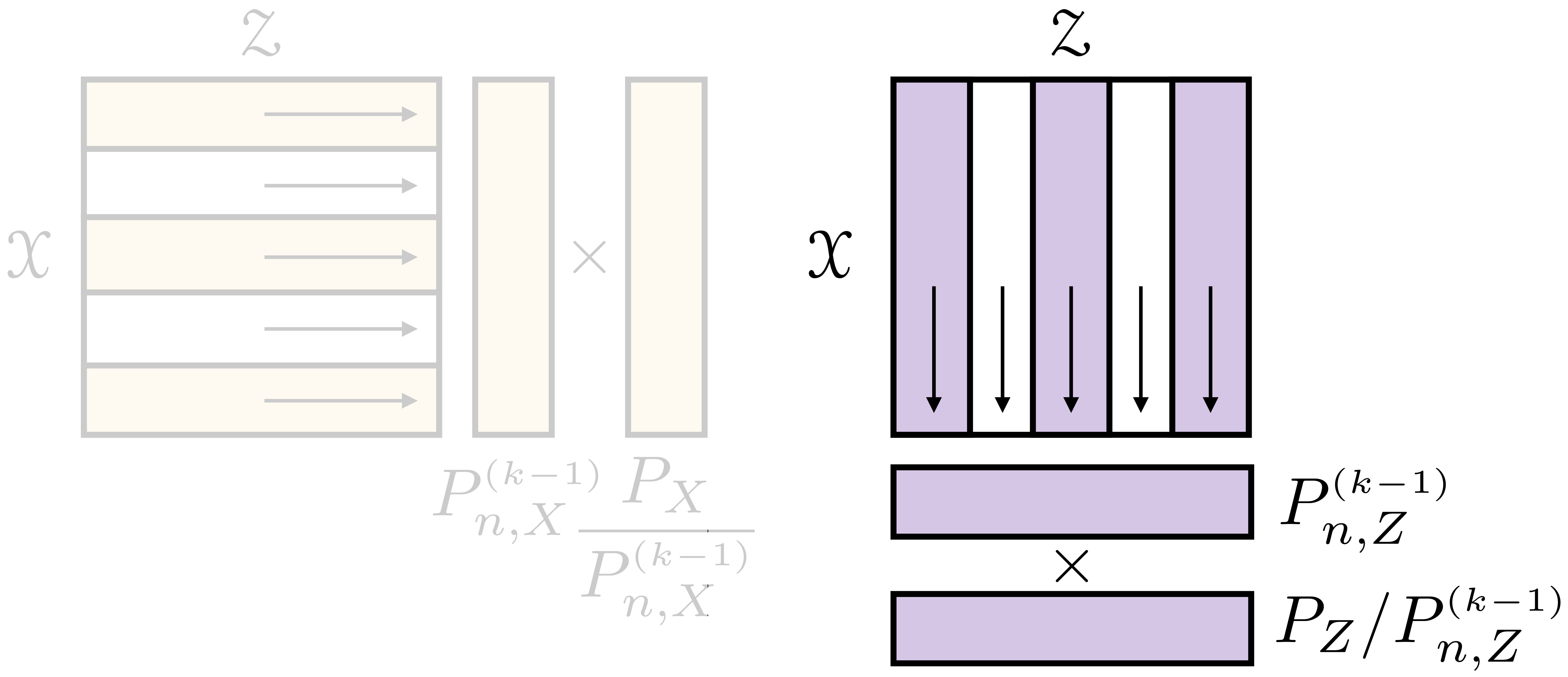
Even Iterations

$$P_n^{(k-1)} \mapsto \frac{P_Z}{P_{n,Z}^{(k-1)}} \otimes P_n^{(k-1)}$$

Marginals are incorporated by **data balancing.**

(Sinkhorn Iterations, Iterative Proportional Fitting, Raking Ratio Estimation)

$P_n^{(k-1)}$



Odd Iterations

Even Iterations

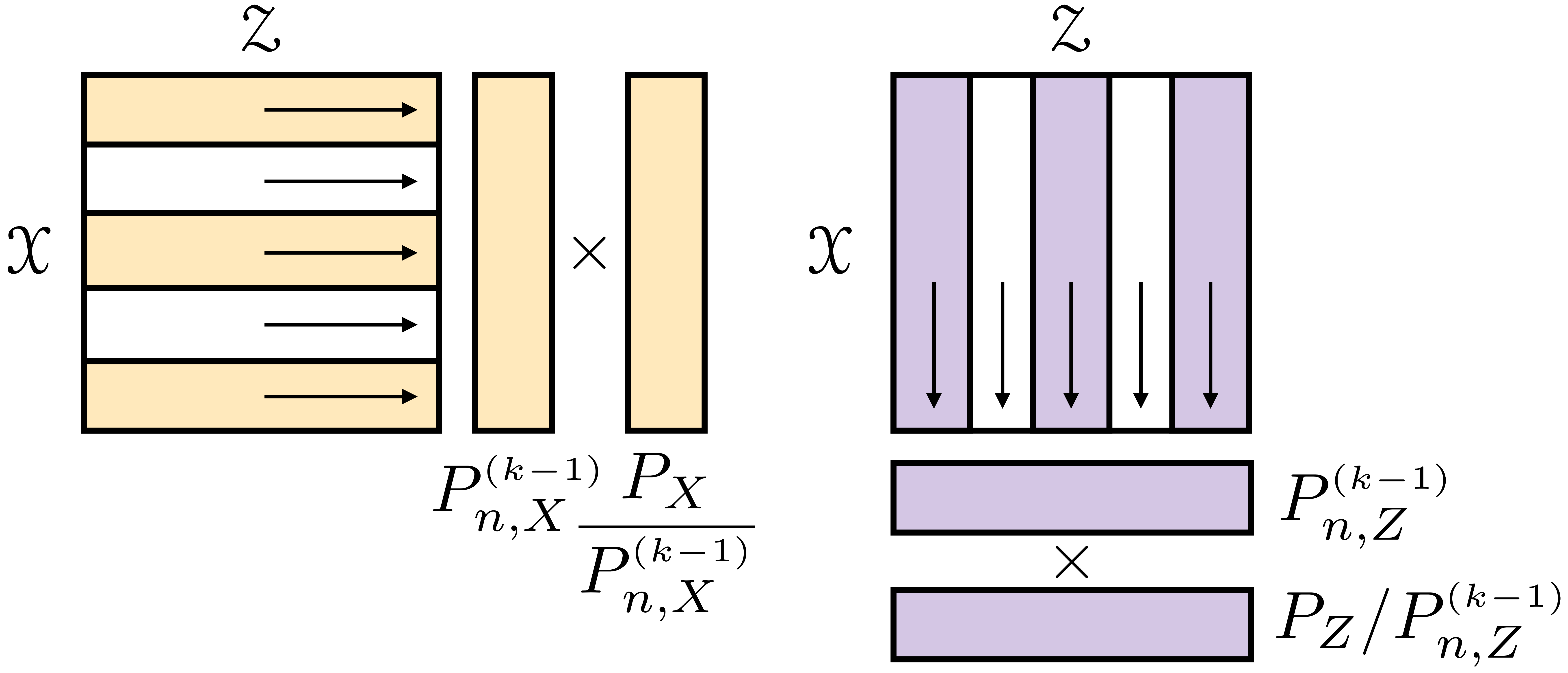
$$P_n^{(k-1)} \mapsto \frac{P_X}{P_{n,X}^{(k-1)}} \otimes P_n^{(k-1)}$$

$$P_n^{(k-1)} \mapsto \frac{P_Z}{P_{n,Z}^{(k-1)}} \otimes P_n^{(k-1)}$$

Marginals are incorporated by **data balancing.**

(Sinkhorn Iterations, Iterative Proportional Fitting, Raking Ratio Estimation)

$P_n^{(k-1)}$



Odd Iterations

Even Iterations

$$P_n^{(k-1)} \mapsto \frac{P_X}{P_{n,X}^{(k-1)}} \otimes P_n^{(k-1)} \qquad P_n^{(k-1)} \mapsto \frac{P_Z}{P_{n,Z}^{(k-1)}} \otimes P_n^{(k-1)}$$

Contributions. We show that:

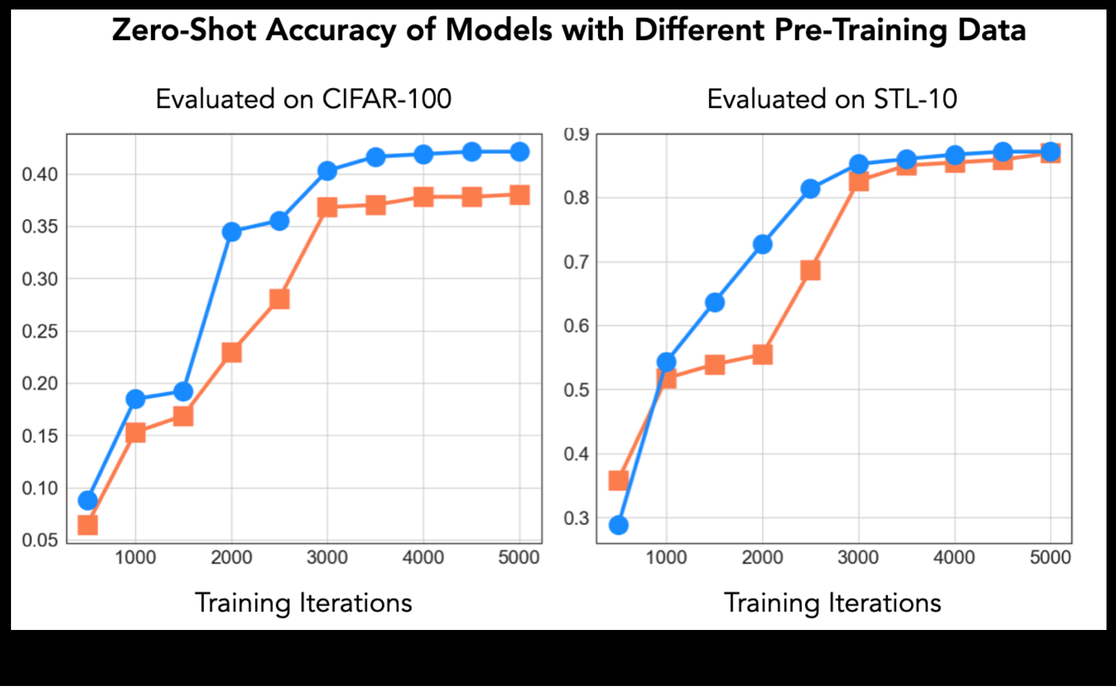
The data curation procedure used in CLIP is an instance of balancing at the **pre-training set scale**.

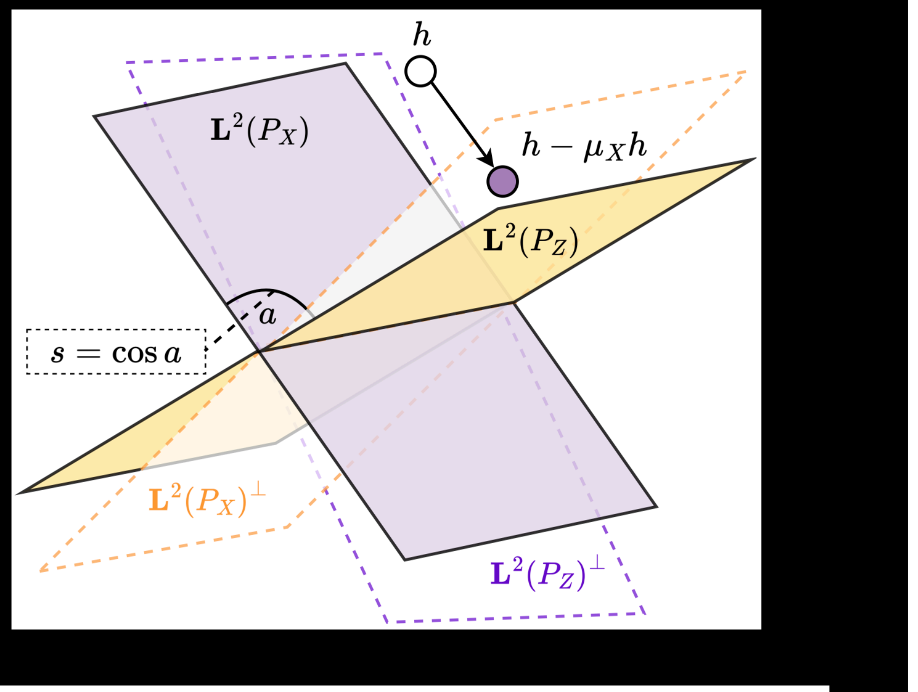
The CLIP objective computes a functional balanced probability measure at the **mini-batch scale**.

Contributions. We show that:

The data curation procedure used in CLIP is an instance of balancing at the **pre-training set scale**.

We quantify the theoretical improvement of using such a procedure in terms of variance-reduced estimation of the population loss.


Zero-Shot Accuracy of Models with Different Pre-Training Data

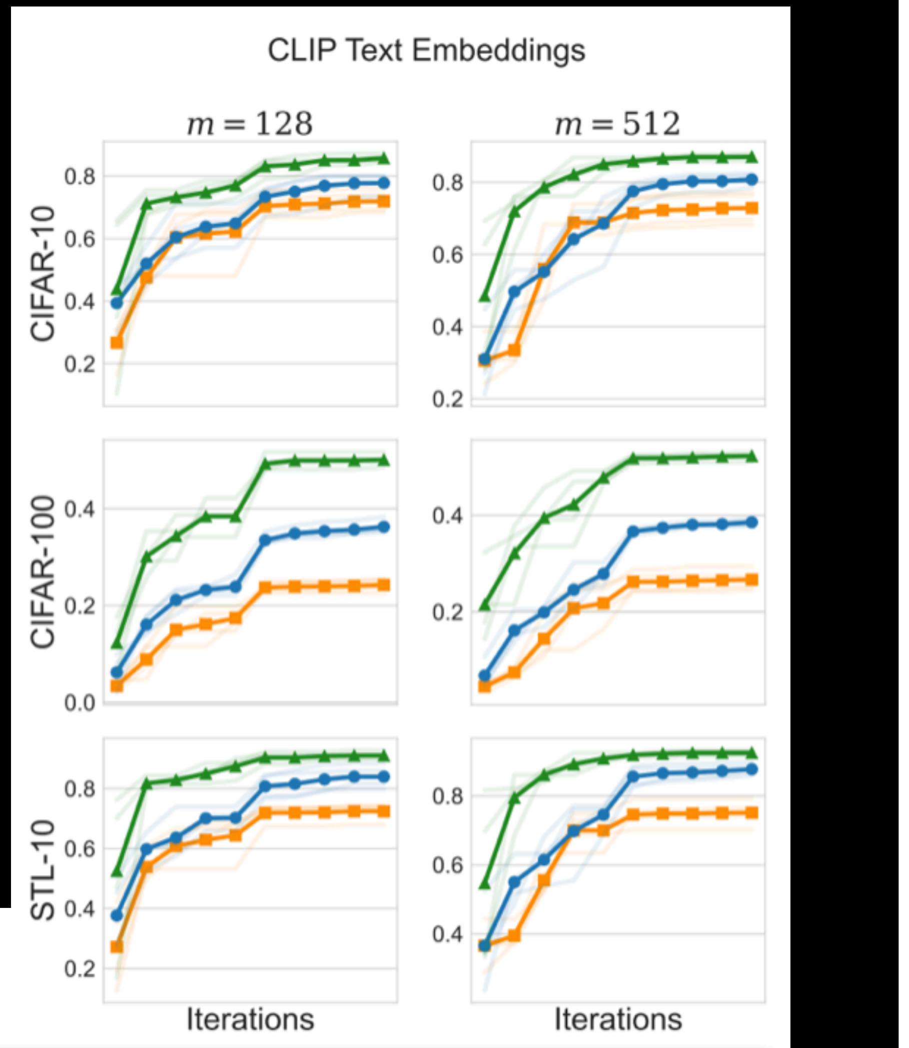


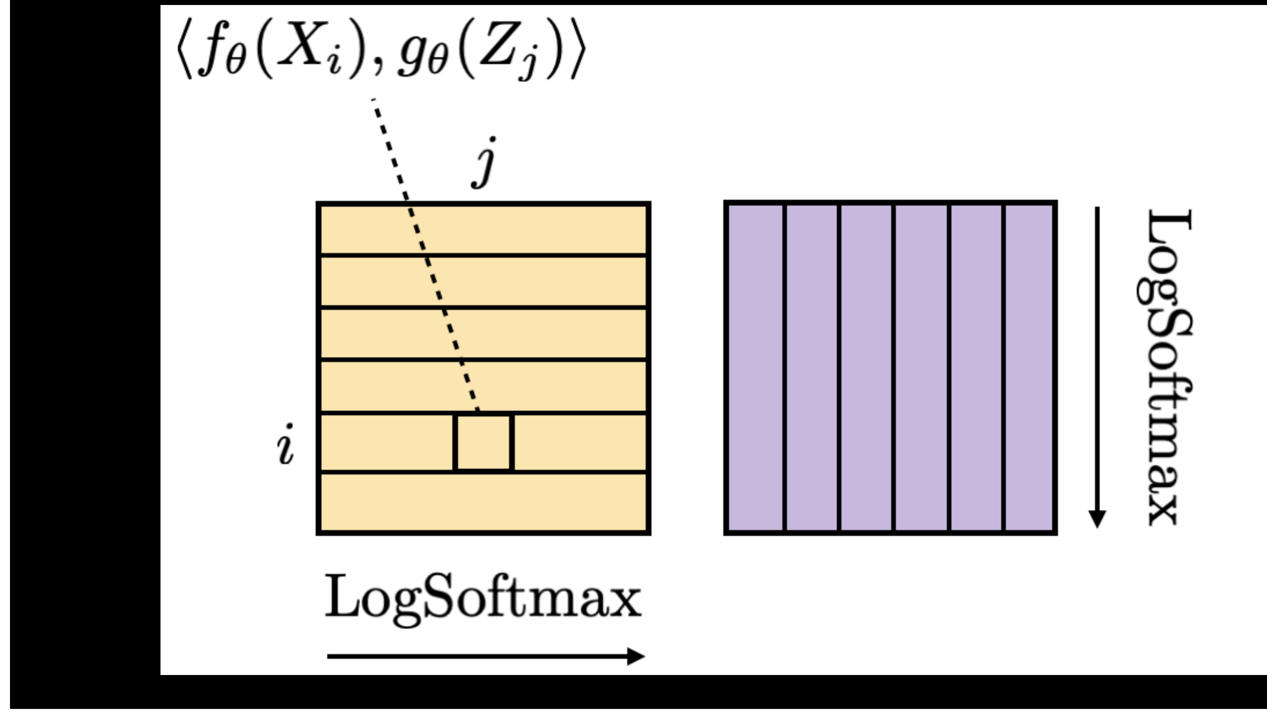
Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[(P_n^{(k)}(h) - P(h))^2\right] = \frac{\mathrm{Var}(\overbrace{\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X}^{k \text{ times}} h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$

The CLIP objective computes a functional balanced probability measure at the **mini-batch scale**.

We use this viewpoint to propose an alternative CLIP-like objective that improves zero-shot classification performance empirically.


CLIP Text Embeddings



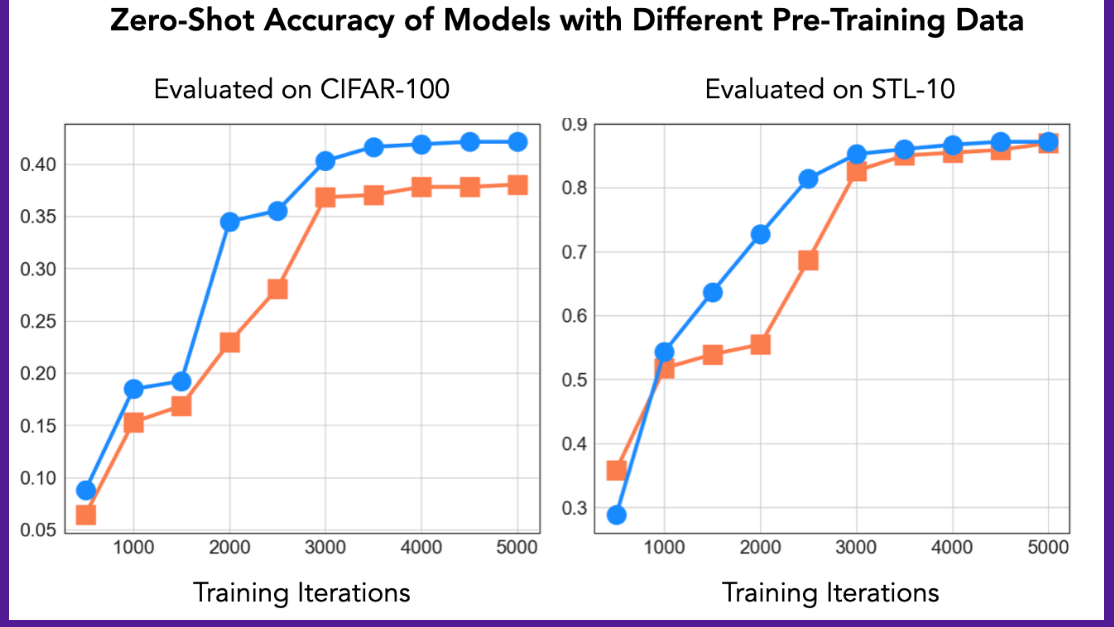
```python
def clip_loss(logits):
    cx   = F.log_softmax(logits, dim=1)
    cy   = F.log_softmax(logits, dim=0)
    return -torch.mean(0.5 * torch.diagonal(cx) + 0.5 * torch.diagonal(cy))
```

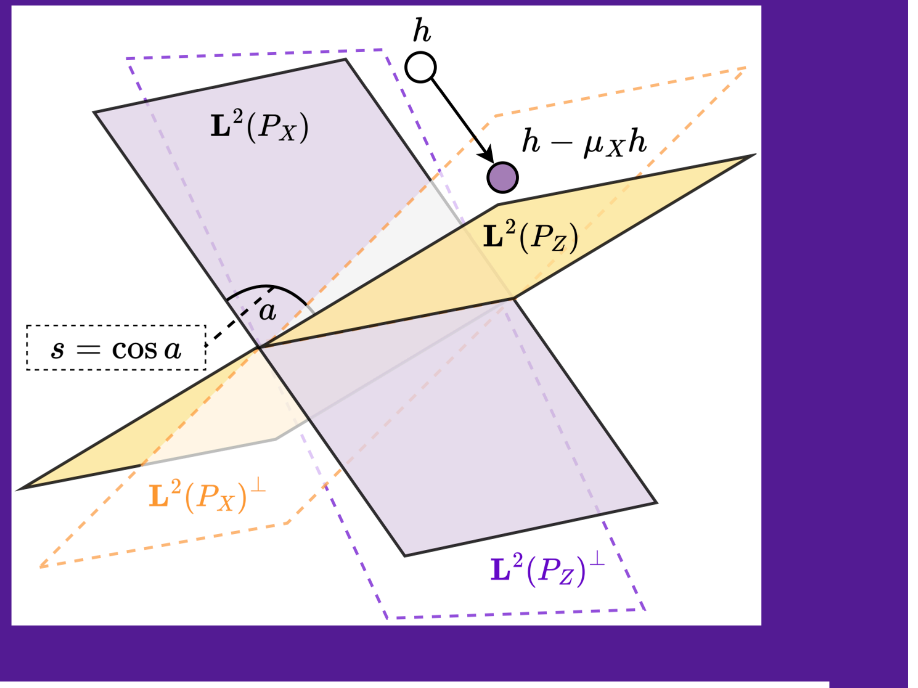
```python
def doubly_centered_loss(logits):
    cx   = F.log_softmax(logits, dim=1)
    cy   = F.log_softmax(logits, dim=0)
    cycx = F.log_softmax(cx, dim=0)
    cxcy = F.log_softmax(cy, dim=1)
    return -torch.mean(0.5 * torch.diagonal(cycx) + 0.5 * torch.diagonal(cxcy))
```

Contributions. We show that:

The data curation procedure used in CLIP is an instance of balancing at the **pre-training set scale**.

We quantify the theoretical improvement of using such a procedure in terms of variance-reduced estimation of the population loss.



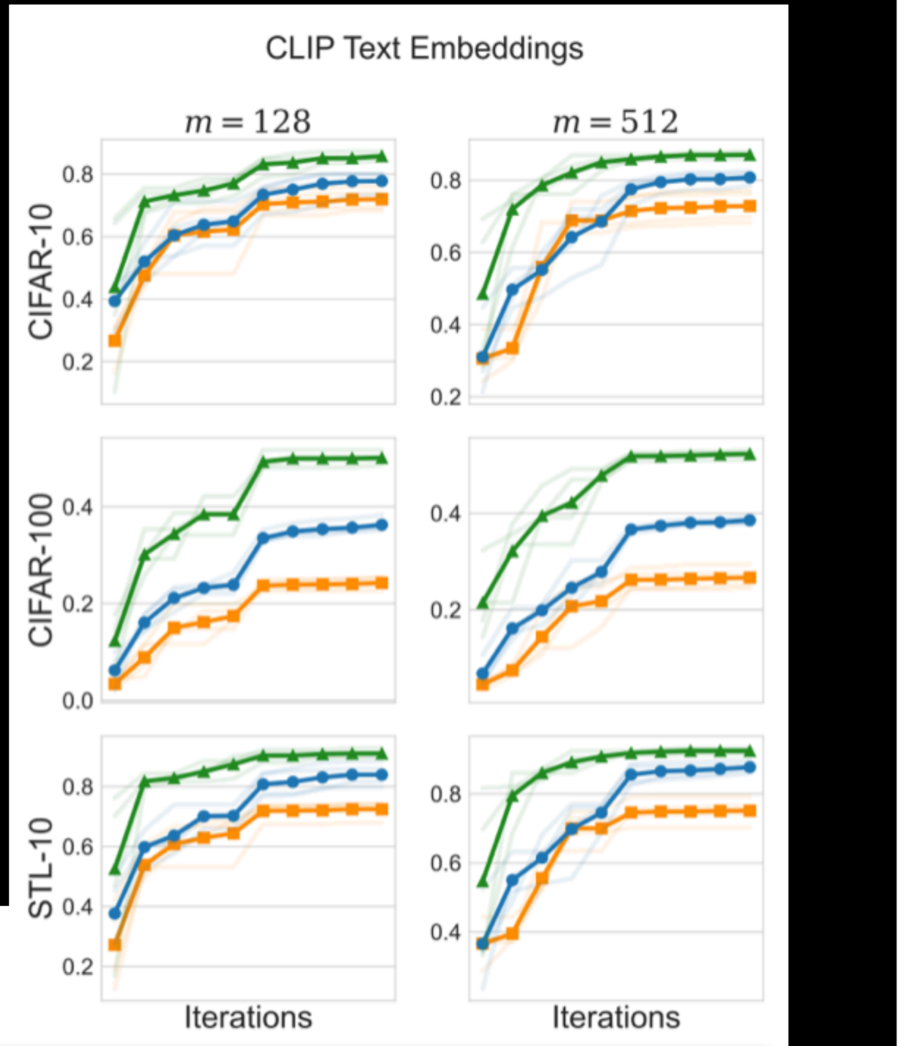


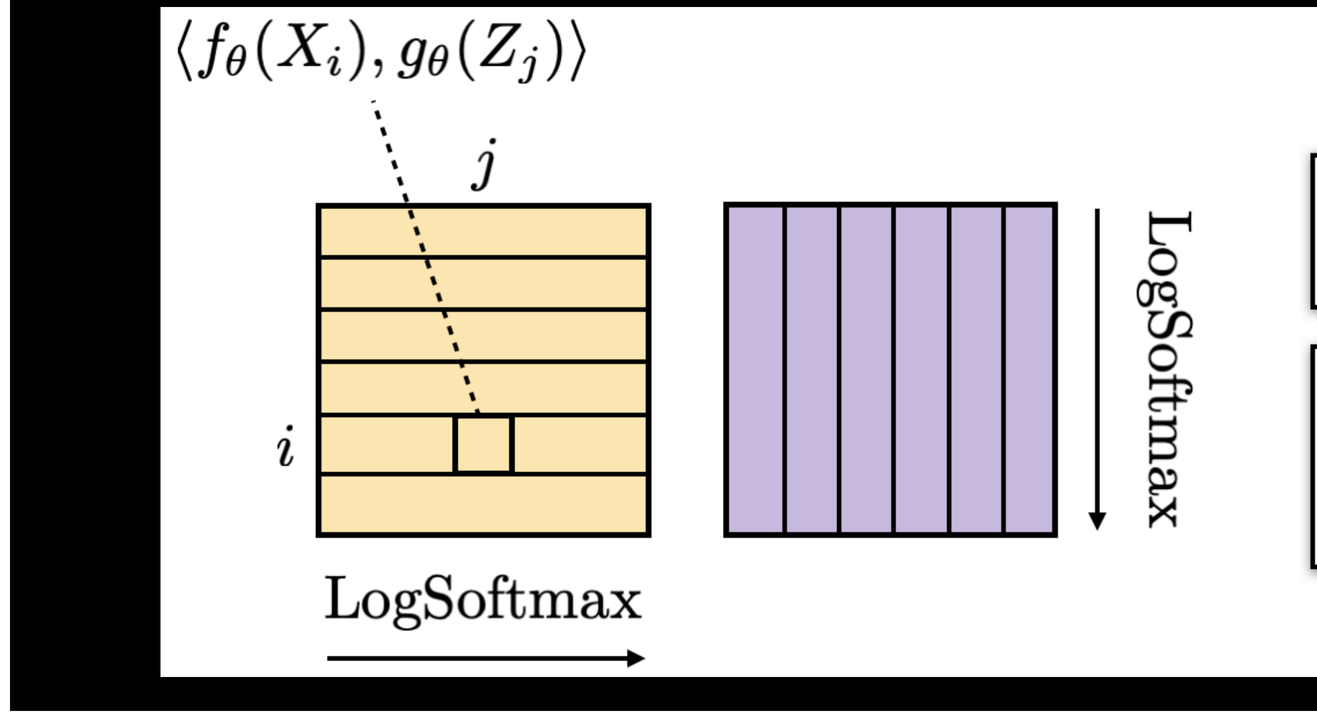
Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[(P_n^{(k)}(h) - P(h))^2\right] = \frac{\mathrm{Var}(\overbrace{\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X}^{k \text{ times}} h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$

The CLIP objective computes a functional balanced probability measure at the **mini-batch scale**.

We use this viewpoint to propose an alternative CLIP-like objective that improves zero-shot classification performance empirically.

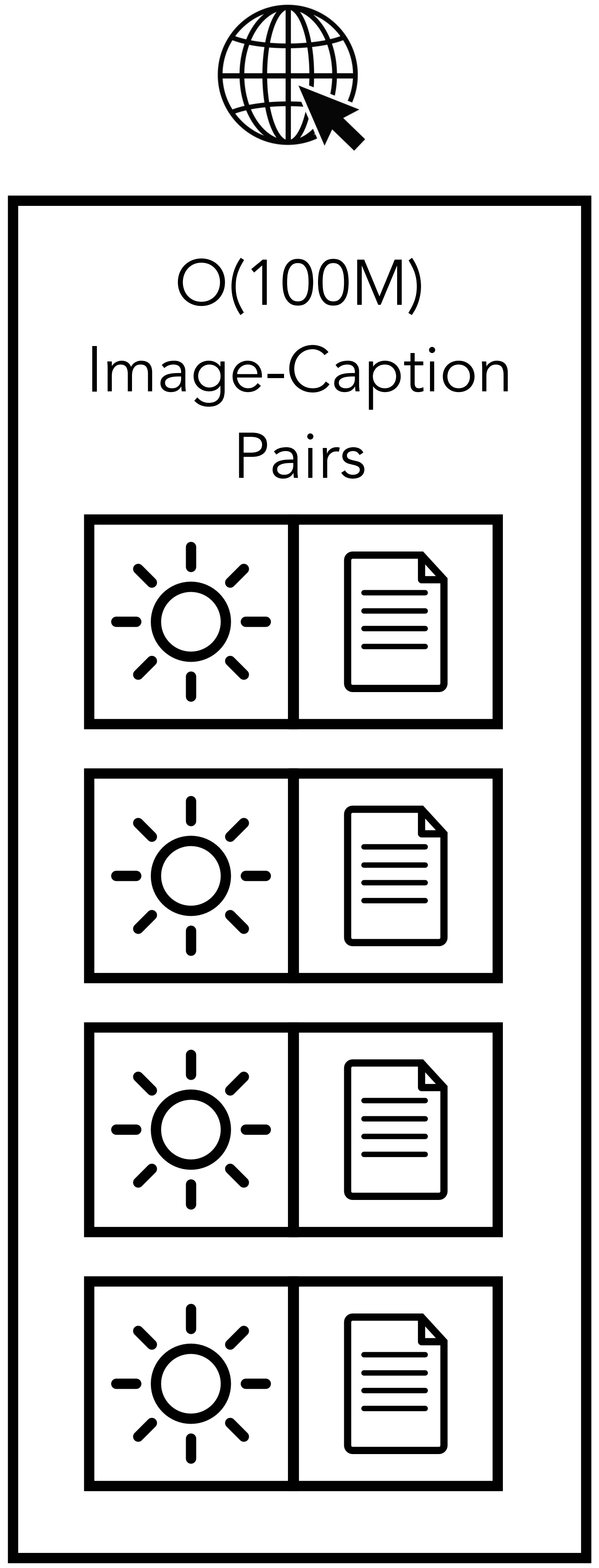




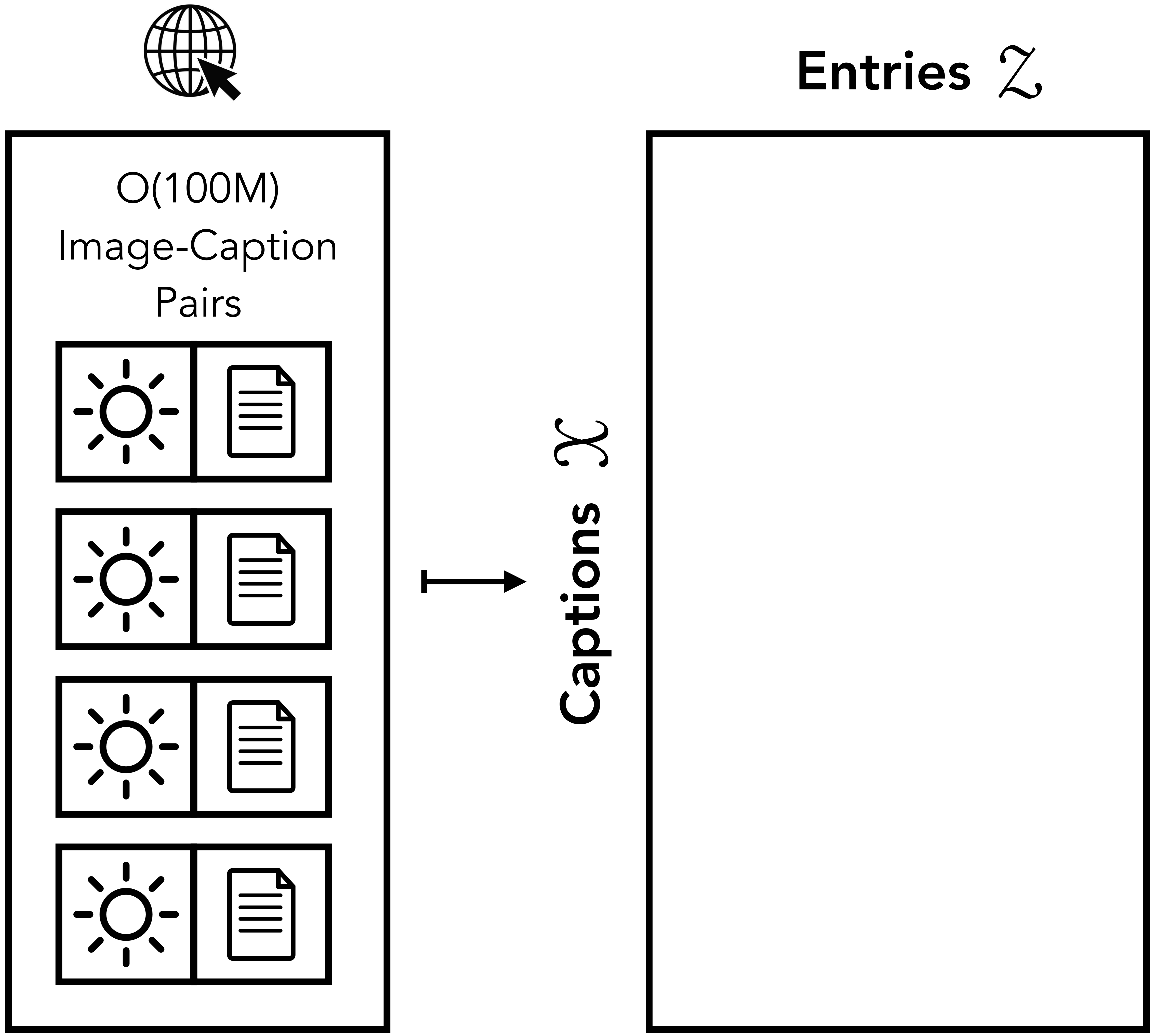
```
def clip_loss(logits):
    cx   = F.log_softmax(logits, dim=1)
    cy   = F.log_softmax(logits, dim=0)
    return -torch.mean(0.5 * torch.diagonal(cx) + 0.5 * torch.diagonal(cy))
```

```
def doubly_centered_loss(logits):
    cx   = F.log_softmax(logits, dim=1)
    cy   = F.log_softmax(logits, dim=0)
    cycx = F.log_softmax(cx, dim=0)
    cxcy = F.log_softmax(cy, dim=1)
    return -torch.mean(0.5 * torch.diagonal(cycx) + 0.5 * torch.diagonal(cxcy))
```

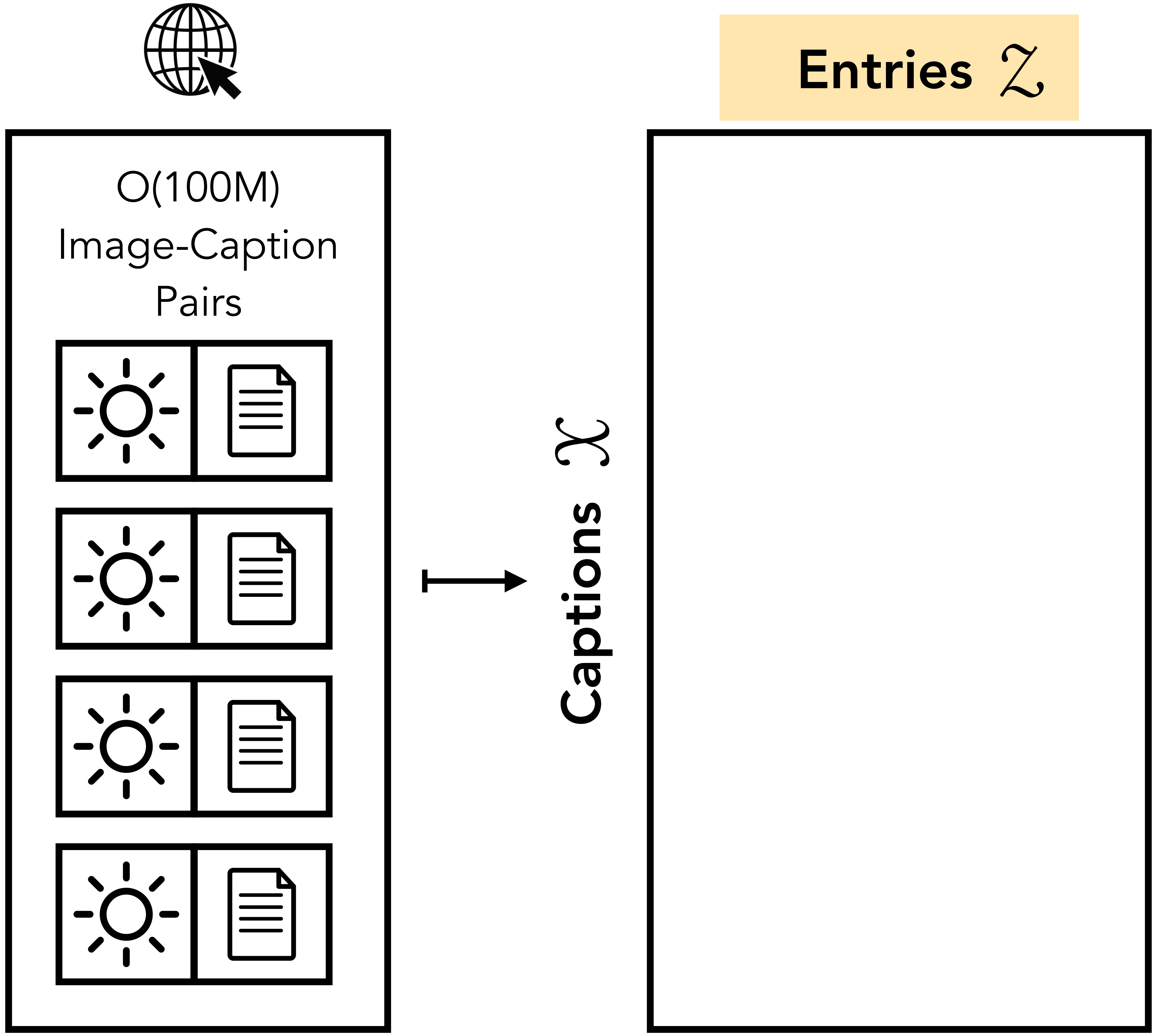
Pre-Training Data Curation: Balancing Keyword Distributions

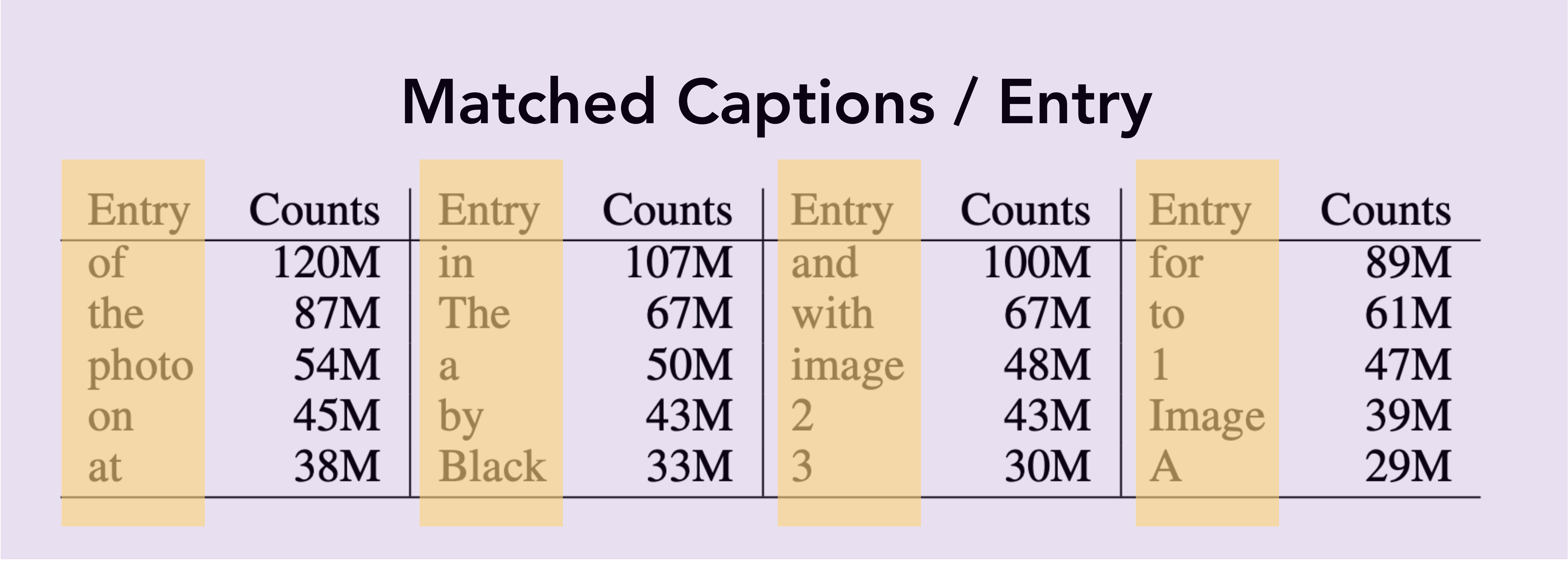


Pre-Training Data Curation: Balancing Keyword Distributions



Pre-Training Data Curation: Balancing Keyword Distributions

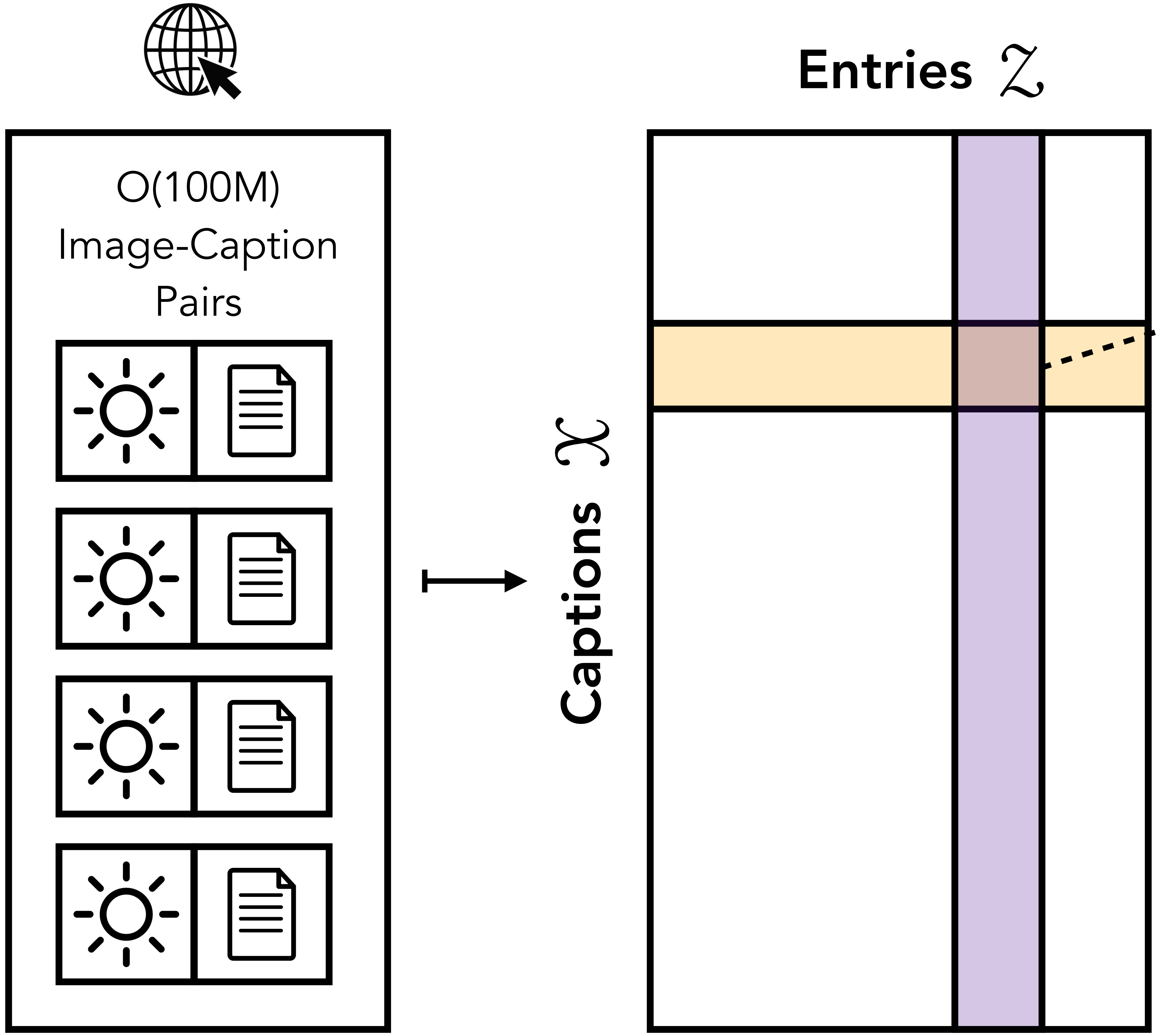




Matched Captions / Entry

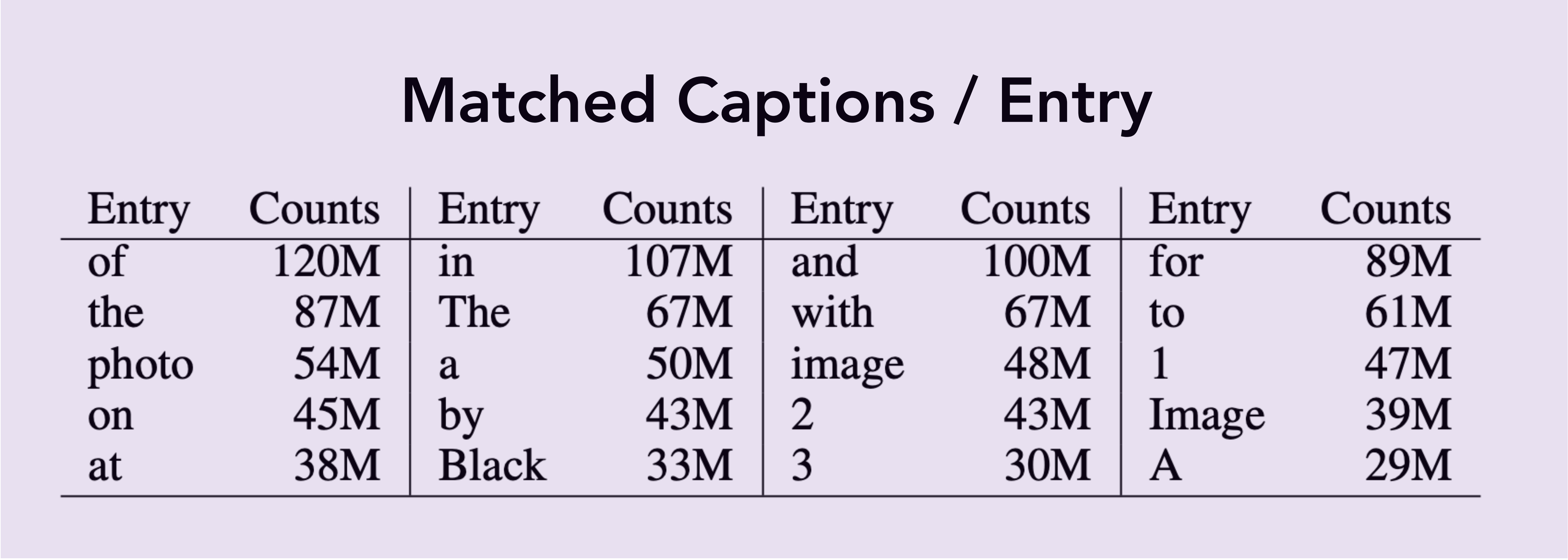
Entry	Counts	Entry	Counts	Entry	Counts	Entry	Counts
of	120M	in	107M	and	100M	for	89M
the	87M	The	67M	with	67M	to	61M
photo	54M	a	50M	image	48M	1	47M
on	45M	by	43M	2	43M	Image	39M
at	38M	Black	33M	3	30M	A	29M

Pre-Training Data Curation: Balancing Keyword Distributions



Entry: photo

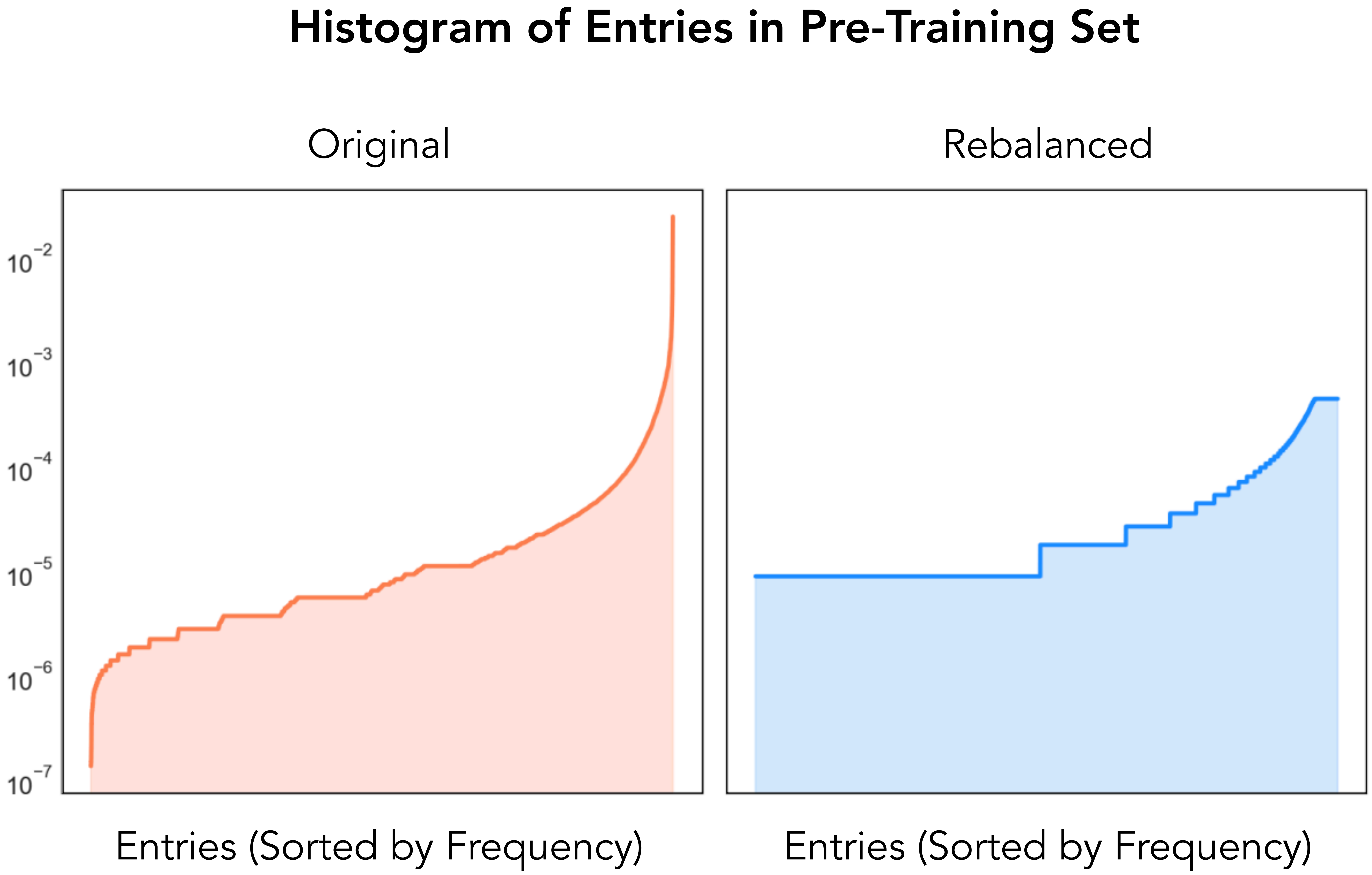
Caption: "photo of a cat"



Matched Captions / Entry

Entry	Counts	Entry	Counts	Entry	Counts	Entry	Counts
of	120M	in	107M	and	100M	for	89M
the	87M	The	67M	with	67M	to	61M
photo	54M	a	50M	image	48M	1	47M
on	45M	by	43M	2	43M	Image	39M
at	38M	Black	33M	3	30M	A	29M

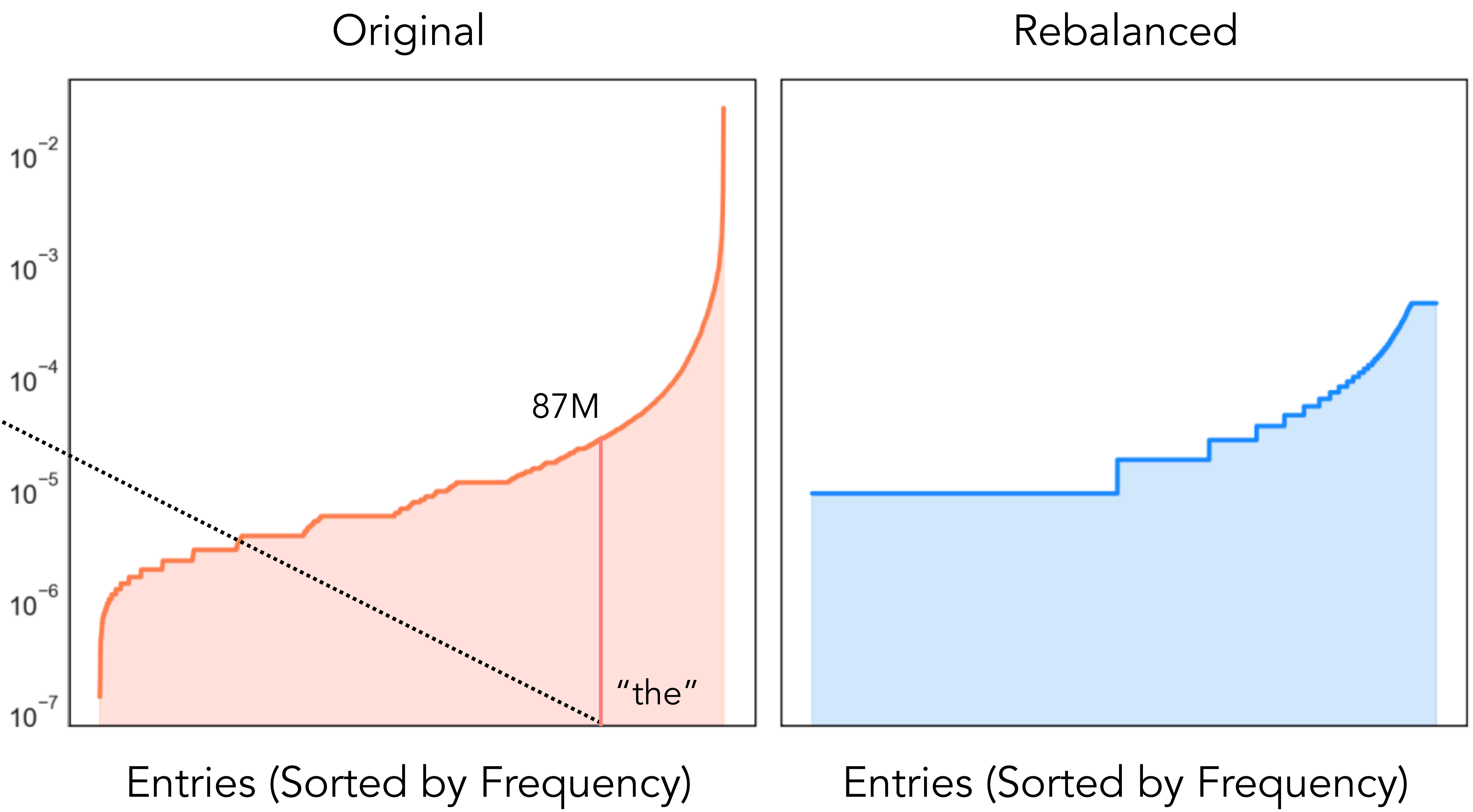
Pre-Training Data Curation: Balancing Keyword Distributions



Pre-Training Data Curation: Balancing Keyword Distributions

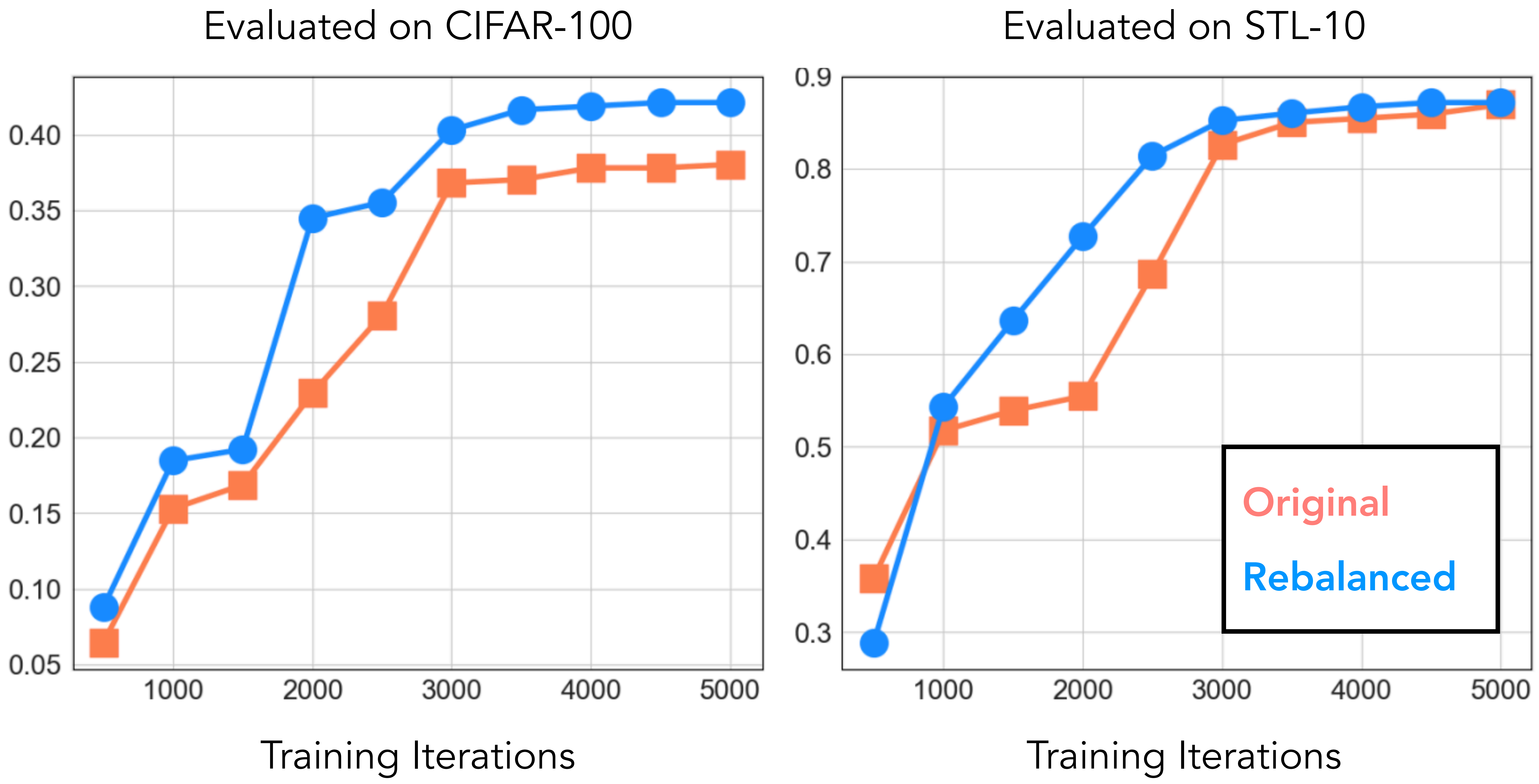
Histogram of Entries in Pre-Training Set

Entry	Counts
of	120M
the	87M
photo	54M
on	45M
at	38M



Pre-Training Data Curation: Balancing Keyword Distributions

Zero-Shot Accuracy of Models with Different Pre-Training Data



Pre-Training Data Curation: Balancing Keyword Distributions

How should we interpret this empirically effective procedure theoretically?

Empirical Risk Minimization with **Marginal Rebalancing**

ERM

$$\min_{\theta \in \mathbb{R}^d} \mathbb{E}_{P_n} \left[h_\theta(X, Z) \right]$$

$\longmapsto$

Rebalanced ERM

$$\min_{\theta \in \mathbb{R}^d} \mathbb{E}_{P_n^{(k)}} \left[h_\theta(X, Z) \right]$$

Empirical Risk Minimization with **Marginal Rebalancing**

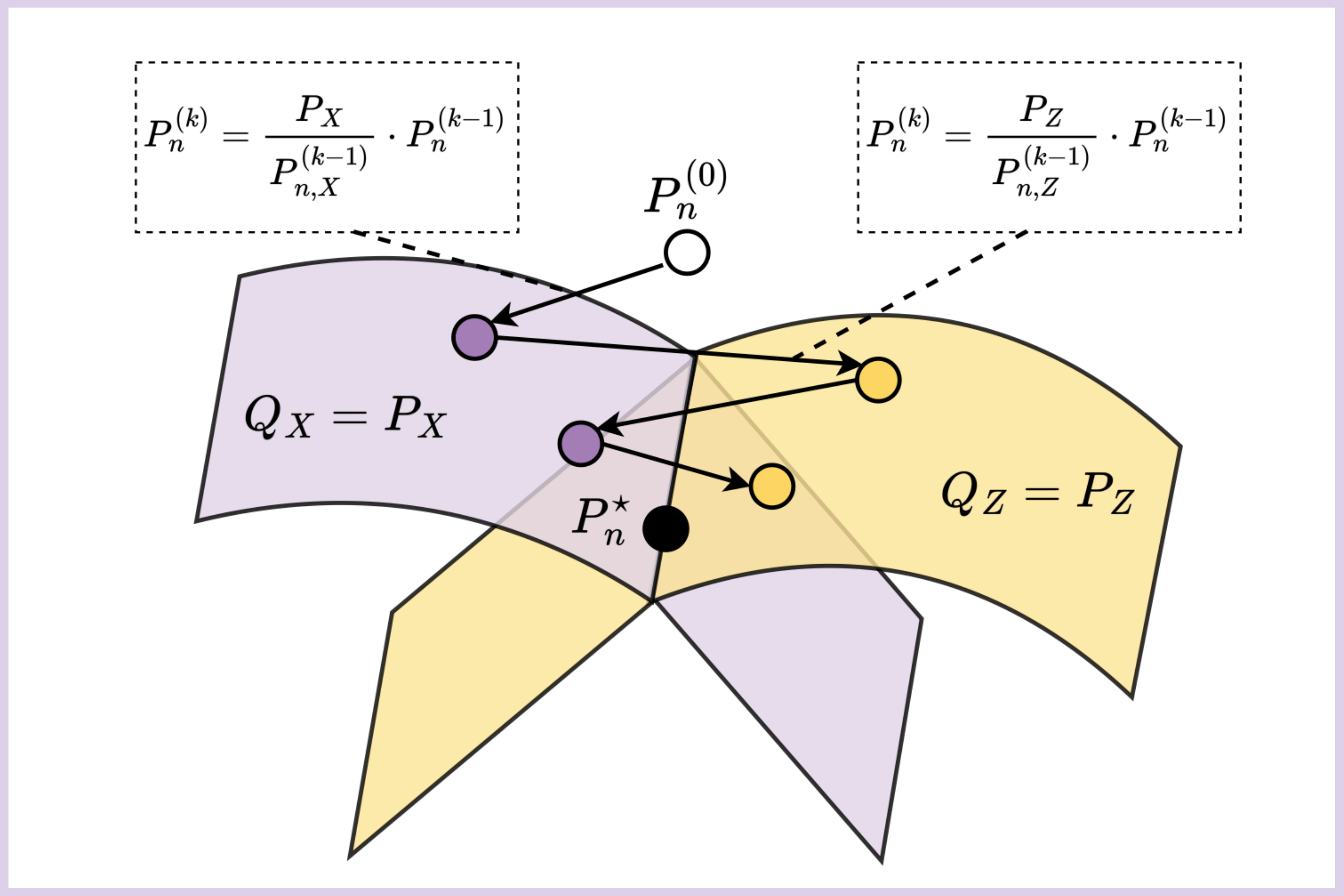
ERM

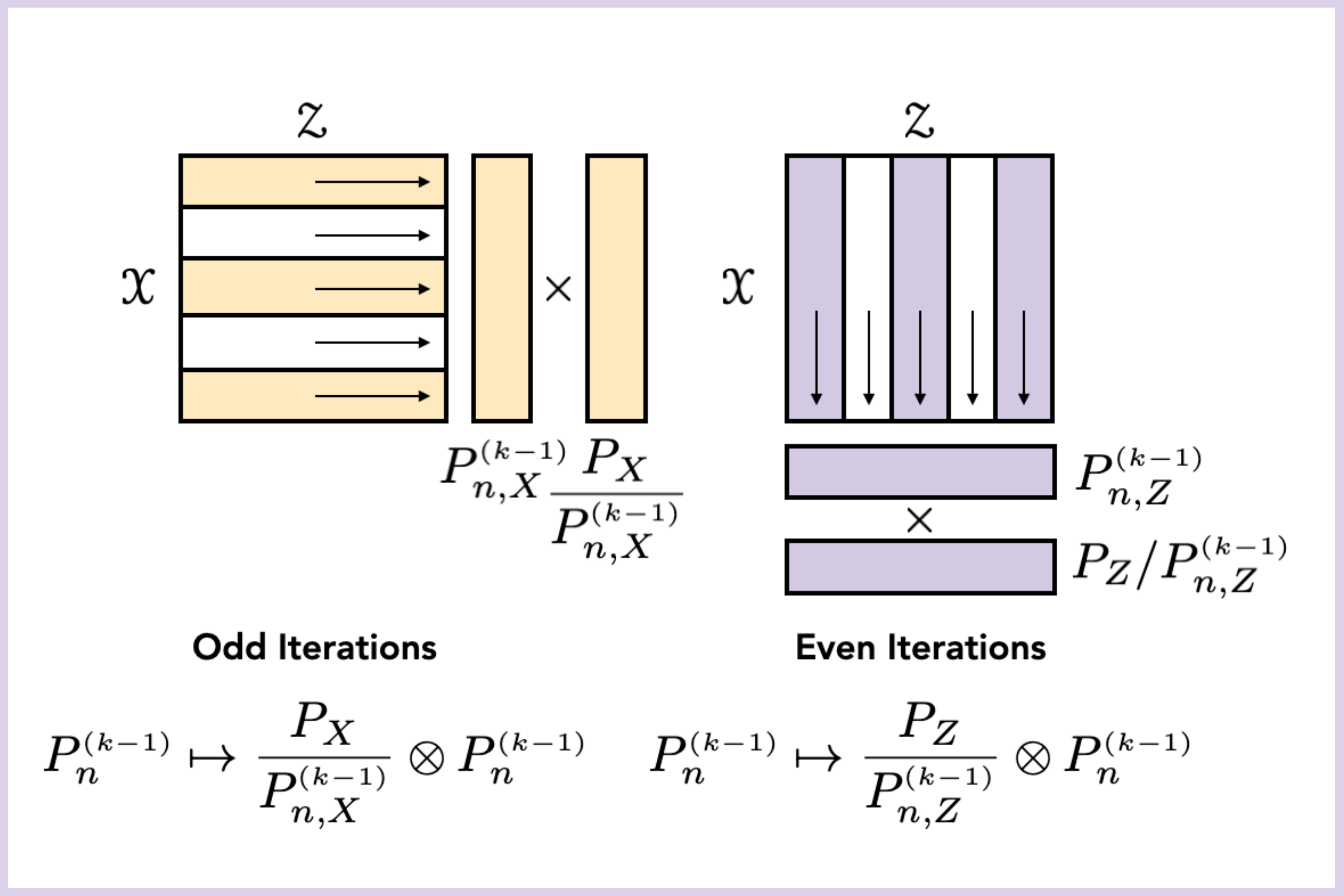
$$\min_{\theta \in \mathbb{R}^d} \mathbb{E}_{P_n} [h_\theta(X, Z)]$$

$\longmapsto$

Rebalanced ERM

$$\min_{\theta \in \mathbb{R}^d} \mathbb{E}_{P_n^{(k)}} [h_\theta(X, Z)]$$


$P_n^{(k)} = \frac{P_X}{P_{n,X}^{(k-1)}} \cdot P_n^{(k-1)}$
$P_n^{(k)} = \frac{P_Z}{P_{n,Z}^{(k-1)}} \cdot P_n^{(k-1)}$
$P_n^{(0)}$
$Q_X = P_X$
$P_n^\star$
$Q_Z = P_Z$


$\mathcal{Z}$
$\mathcal{X}$
$\times$
$P_{n,X}^{(k-1)}$
$\frac{P_X}{P_{n,X}^{(k-1)}}$
$\mathcal{Z}$
$\mathcal{X}$
$P_{n,Z}^{(k-1)}$
$\times$
$P_Z/P_{n,Z}^{(k-1)}$
Odd Iterations
Even Iterations
$P_n^{(k-1)} \mapsto \frac{P_X}{P_{n,X}^{(k-1)}} \otimes P_n^{(k-1)}$
$P_n^{(k-1)} \mapsto \frac{P_Z}{P_{n,Z}^{(k-1)}} \otimes P_n^{(k-1)}$

Empirical Risk Minimization with **Marginal Rebalancing**

ERM

$$\min_{\theta\in\mathbb{R}^d} \mathbb{E}_{P_n}\left[h_\theta(X,Z)\right]$$

$$\longmapsto$$

Rebalanced ERM

$$\min_{\theta\in\mathbb{R}^d} \underbrace{\mathbb{E}_{P_n^{(k)}}\left[h_\theta(X,Z)\right]}_{= P_n^{(k)}(h) \overset{?}{\approx} P(h)}$$

We hide the dependence on θ and consider point-wise estimation for a fixed $h \equiv h_\theta$.

Empirical Risk Minimization with **Marginal Rebalancing**

ERM

$$\min_{\theta \in \mathbb{R}^d} \mathbb{E}_{P_n}\left[h_\theta(X, Z)\right]$$

$$\longmapsto$$

Rebalanced ERM

$$\min_{\theta \in \mathbb{R}^d} \underbrace{\mathbb{E}_{P_n^{(k)}}\left[h_\theta(X, Z)\right]}_{}$$

$$= P_n^{(k)}(h) \overset{?}{\approx} P(h)$$

We measure the benefit of balancing via **variance/MSE reduction** for estimating the expectation of a fixed test function.

$$\mathbb{E}_{P^n}\left[(P_n^{(k)}(h) - P(h))^2\right] \leq \; ? \; < \frac{\mathbb{V}\mathrm{ar}(h)}{n}$$

The main results depend on particular distribution-dependent operators.

The main results depend on particular distribution-dependent operators.

Conditional **Mean** Operators

$$\mu_X : \mathbf{L}^2(P) \to \mathbf{L}^2(P_X)$$

$$\mu_X h = \mathbb{E}\left[h(\cdot, Z) | X\right]$$

$$\mu_Z : \mathbf{L}^2(P) \to \mathbf{L}^2(P_Z)$$

$$\mu_Z h = \mathbb{E}\left[h(X, \cdot) | Z\right]$$

The main results depend on particular distribution-dependent operators.

Conditional **Mean** Operators

$$\mu_X : \mathbf{L}^2(P) \to \mathbf{L}^2(P_X)$$
$$\mu_X h = \mathbb{E}\left[h(\cdot, Z) | X\right]$$

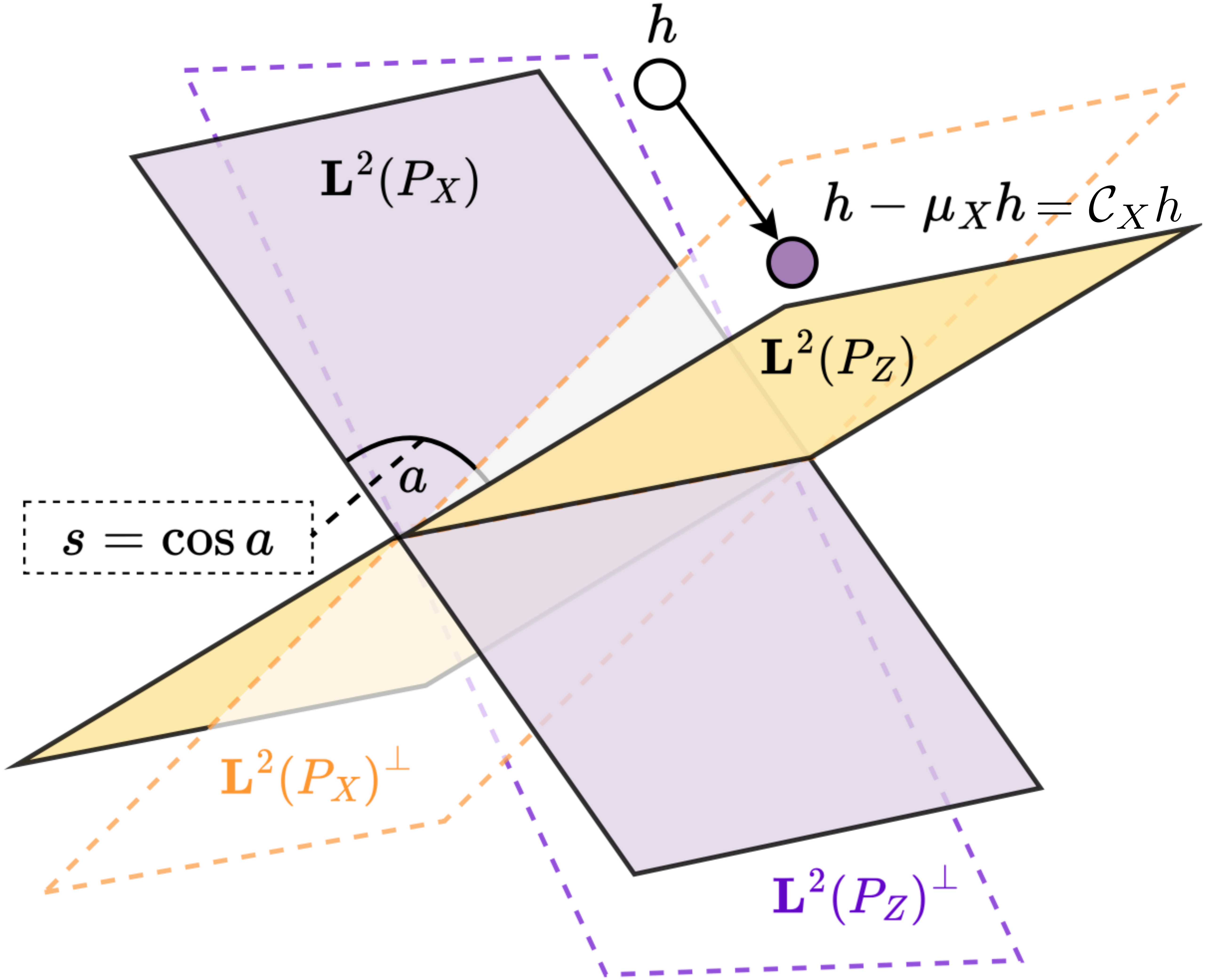
$$\mu_Z : \mathbf{L}^2(P) \to \mathbf{L}^2(P_Z)$$
$$\mu_Z h = \mathbb{E}\left[h(X, \cdot) | Z\right]$$

Conditional **Centering** Operators

$$\mathcal{C}_X : \mathbf{L}^2(P) \to \mathbf{L}^2(P_X)^\perp$$
$$\mathcal{C}_X h = h - \mathbb{E}\left[h(\cdot, Z) | X\right]$$

$$\mathcal{C}_Z : \mathbf{L}^2(P) \to \mathbf{L}^2(P_Z)^\perp$$
$$\mathcal{C}_Z h = h - \mathbb{E}\left[h(X, \cdot) | Z\right]$$

The main results depend on particular distribution-dependent operators.


h
$h - \mu_X h = \mathcal{C}_X h$
$\mathbf{L}^2(P_X)$
$\mathbf{L}^2(P_Z)$
a
$s = \cos a$
$\mathbf{L}^2(P_X)^\perp$
$\mathbf{L}^2(P_Z)^\perp$

The main results depend on particular distribution-dependent operators.

Conditional **Mean** Operators

Projection onto $\mathbf{L}^2(P_X)$

Projection onto $\mathbf{L}^2(P_Z)$

Conditional **Centering** Operators

Projection onto $\mathbf{L}^2(P_X)^\perp$

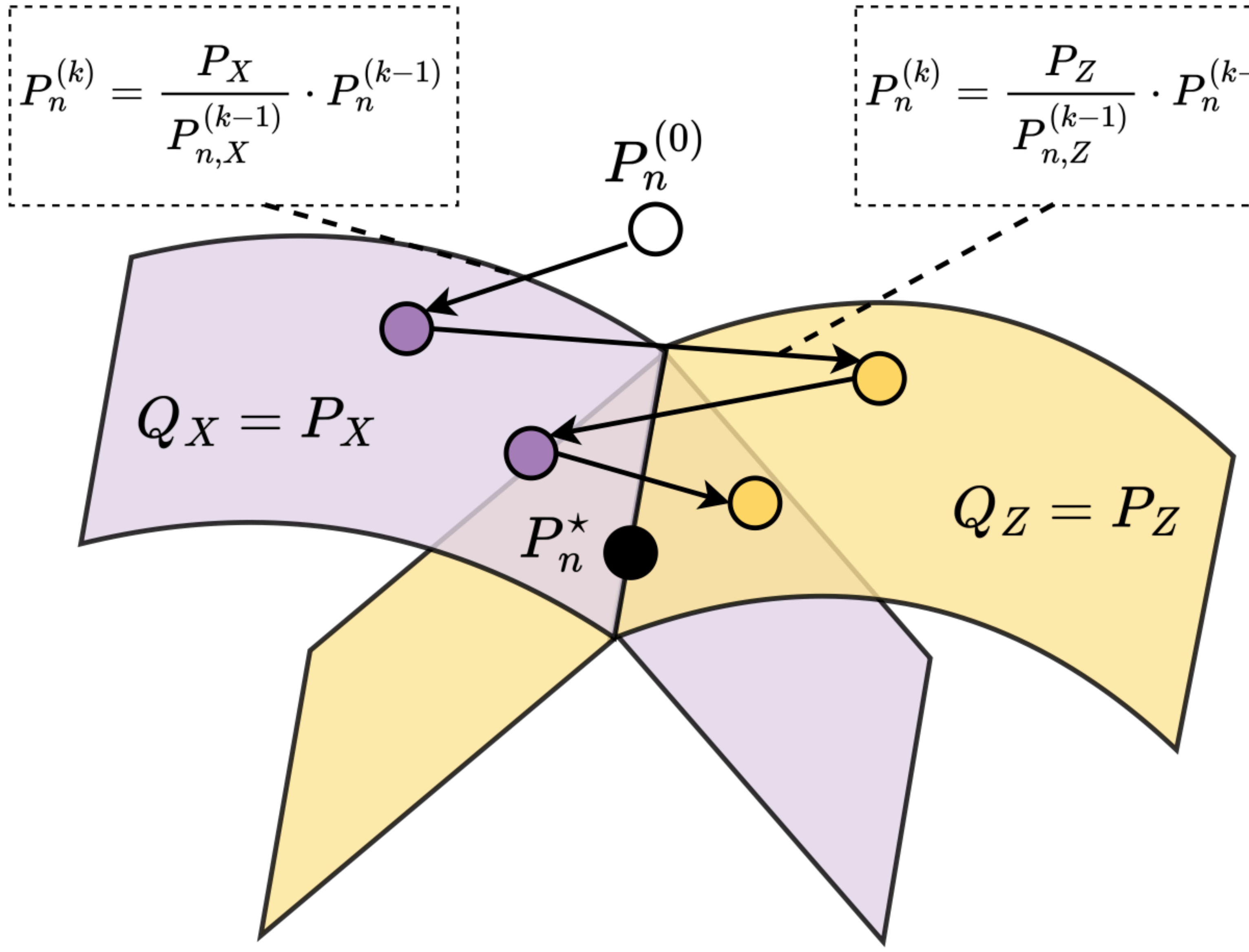
Projection onto $\mathbf{L}^2(P_Z)^\perp$

Theorem (Liu, M., Pal, Harchaoui)

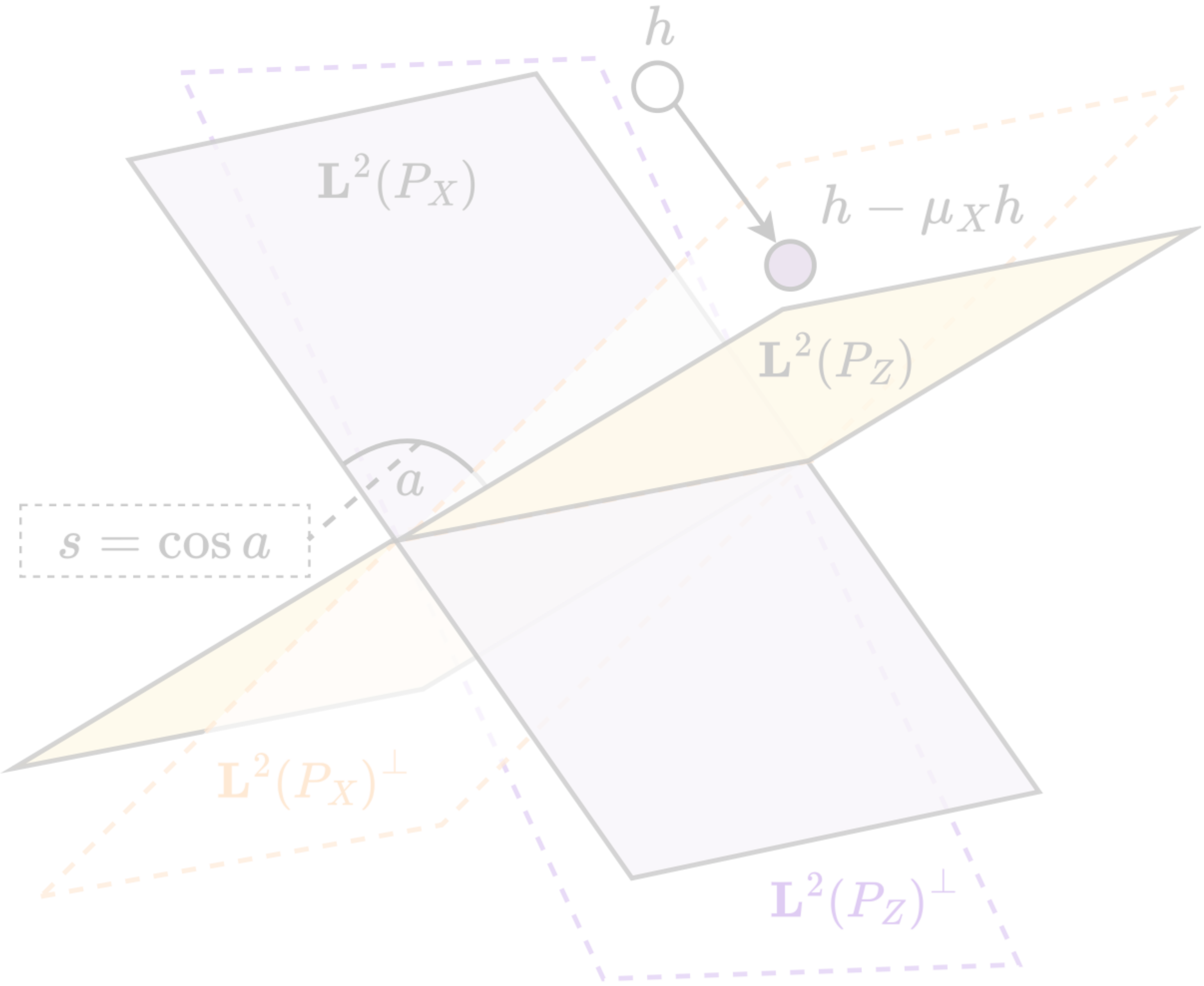
$$\mathbb{E}_{P^n}\left[\left(P_n^{(k)}(h) - P(h)\right)^2\right] = \frac{\mathbb{V}\mathrm{ar}(\overbrace{\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X}^{k \text{ times}} h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$

Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[\left(P_n^{(k)}(h) - P(h)\right)^2\right] = \frac{\mathbb{V}\mathrm{ar}(\overbrace{\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X}^{k \text{ times}} h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$



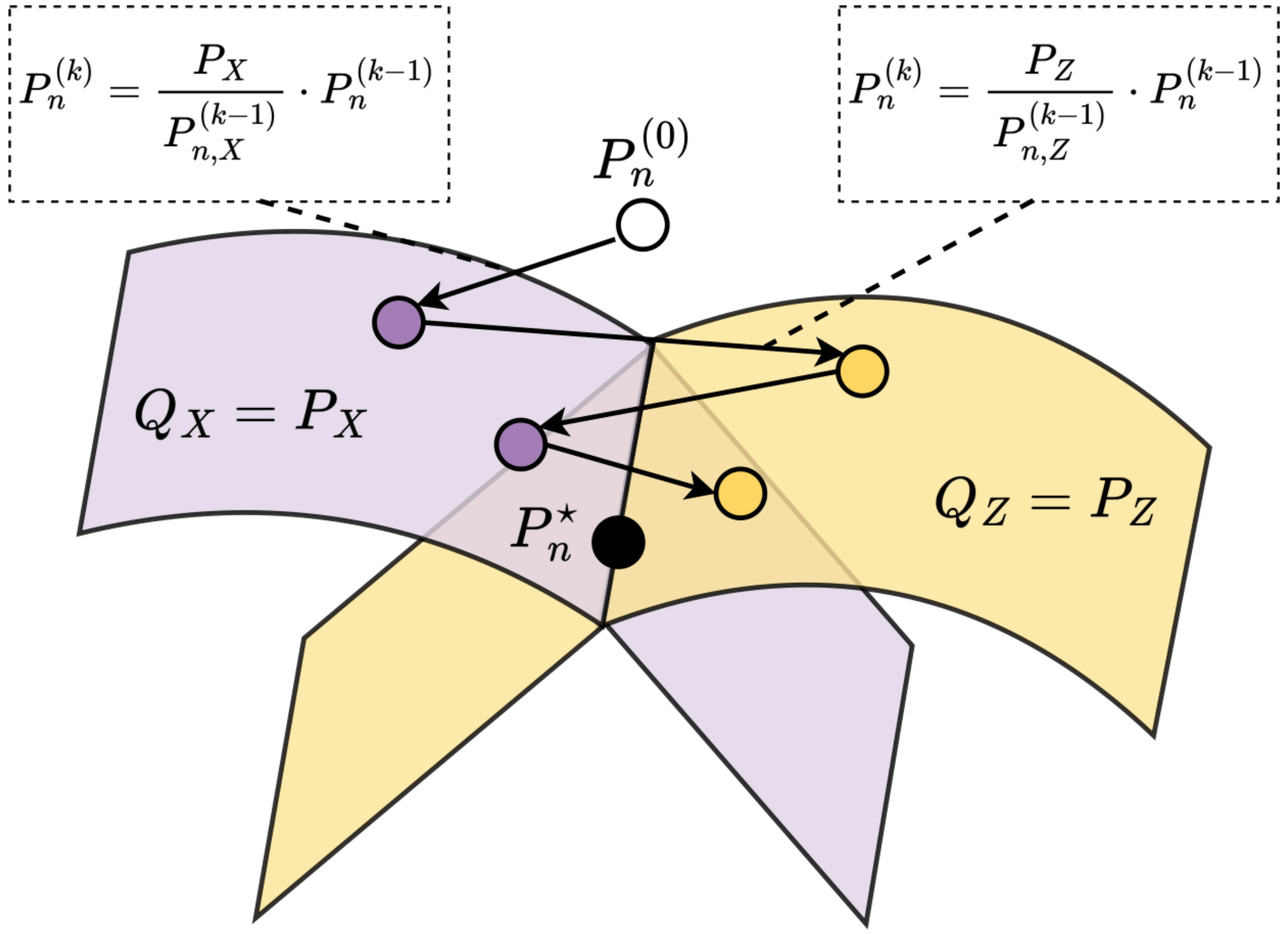
$P_n^{(k)} = \frac{P_X}{P_{n,X}^{(k-1)}} \cdot P_n^{(k-1)}$
$P_n^{(k)} = \frac{P_Z}{P_{n,Z}^{(k-1)}} \cdot P_n^{(k-1)}$
$P_n^{(0)}$
$Q_X = P_X$
$Q_Z = P_Z$
$P_n^\star$



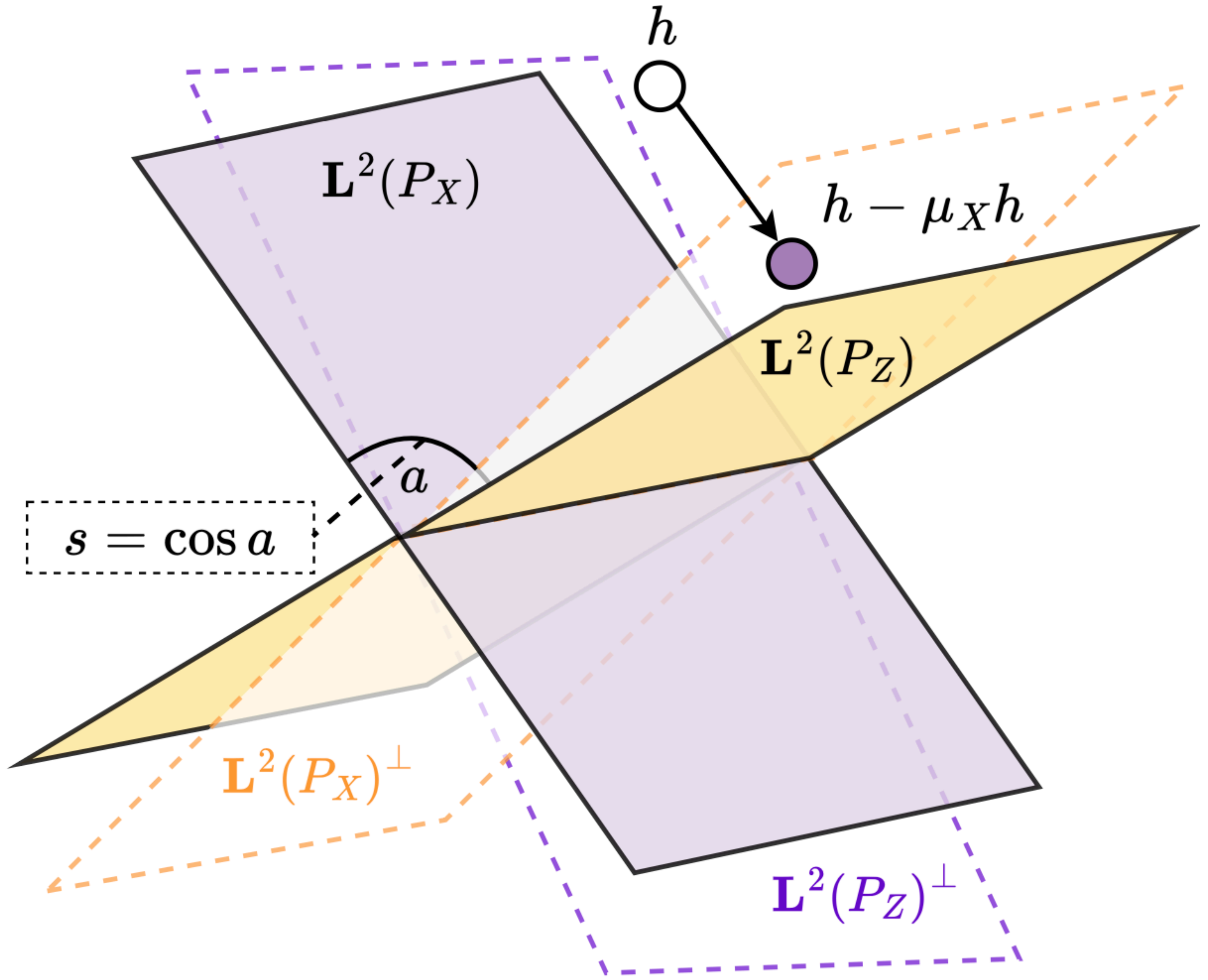
Information Projections ↦ Orthogonal Projections ↦ Variance Reduction

Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[\left(P_n^{(k)}(h) - P(h)\right)^2\right] = \frac{\mathbb{V}\mathrm{ar}(\overbrace{\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X}^{k \text{ times}} h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$



$P_n^{(k)} = \frac{P_X}{P_{n,X}^{(k-1)}} \cdot P_n^{(k-1)}$
$P_n^{(k)} = \frac{P_Z}{P_{n,Z}^{(k-1)}} \cdot P_n^{(k-1)}$
$P_n^{(0)}$
$Q_X = P_X$
$Q_Z = P_Z$
$P_n^\star$

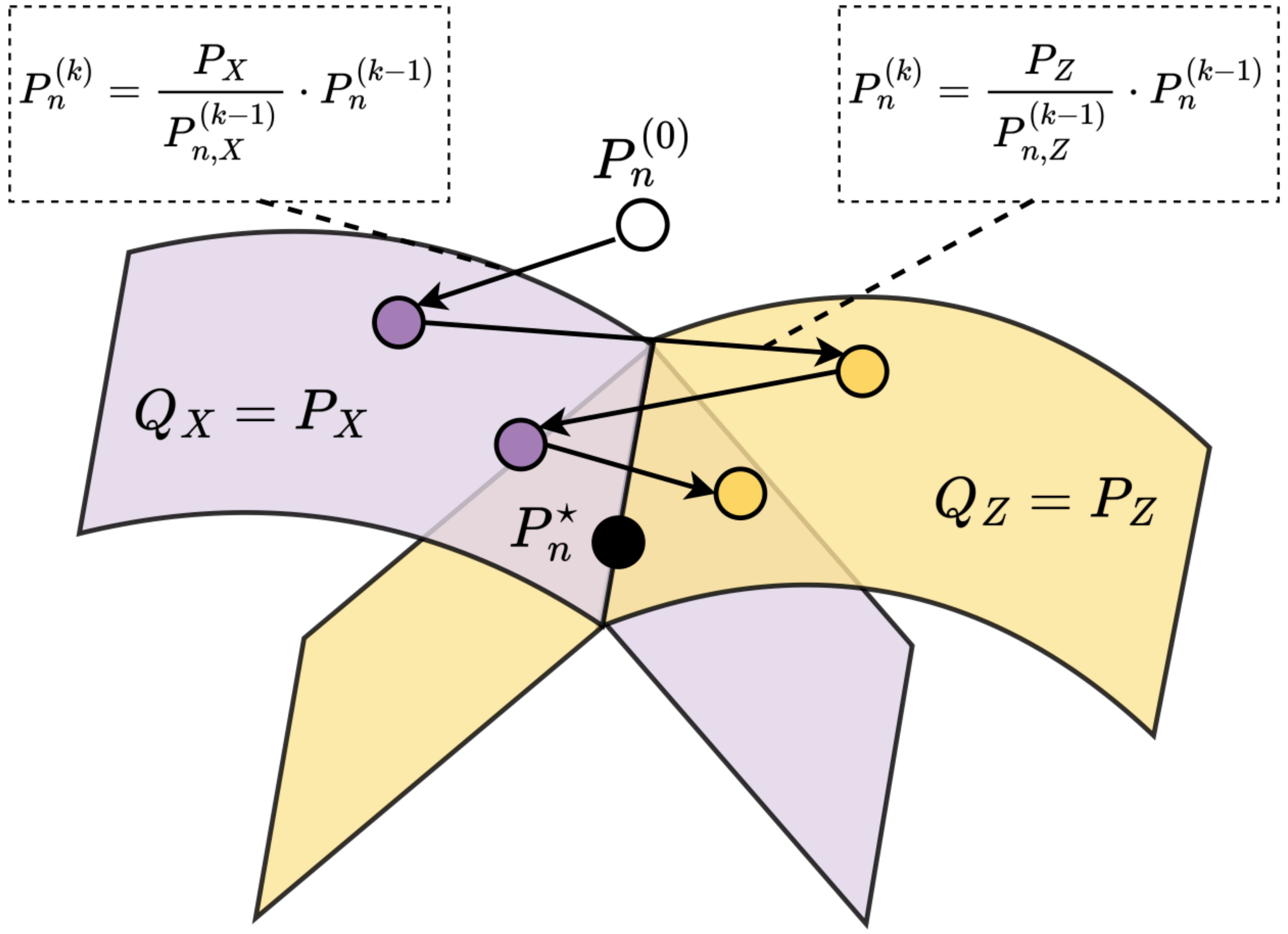


(next slide)

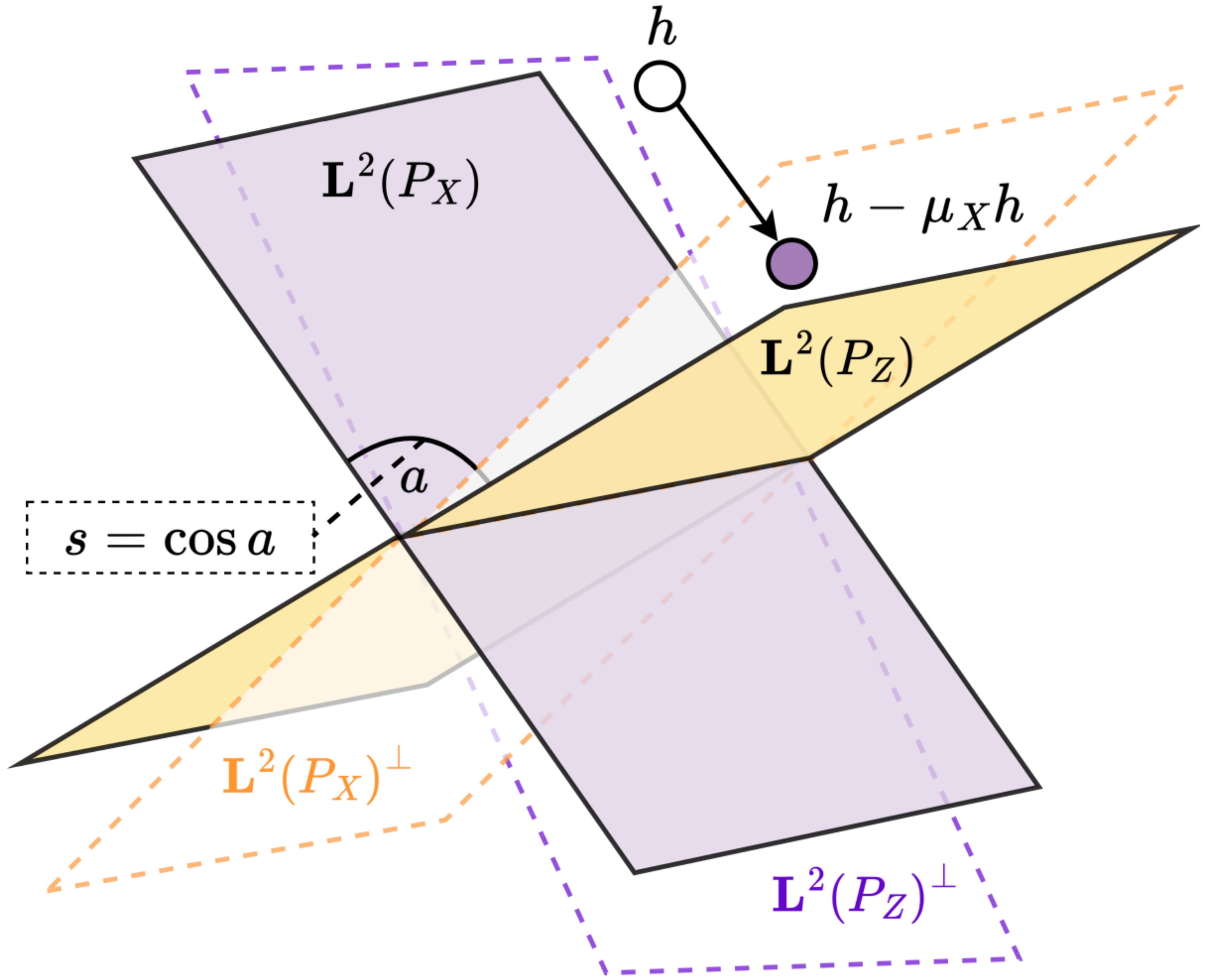
Information Projections ↦ Orthogonal Projections ↦ Variance Reduction

Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[(P_n^{(k)}(h) - P(h))^2\right] = \frac{\mathbb{V}\mathrm{ar}(\overbrace{\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X}^{k \text{ times}} h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$



$P_n^{(k)} = \frac{P_X}{P_{n,X}^{(k-1)}} \cdot P_n^{(k-1)}$
$P_n^{(0)}$
$P_n^{(k)} = \frac{P_Z}{P_{n,Z}^{(k-1)}} \cdot P_n^{(k-1)}$
$Q_X = P_X$
$Q_Z = P_Z$
$P_n^\star$



Information Projections ↦ Orthogonal Projections ↦ Variance Reduction

Proof Technique: Recursive Error Decomposition

Where do these operators come from?

$$(\mu_k, \mathcal{C}_k) := \begin{cases} (\mu_X, \mathcal{C}_X) & k \text{ odd} \\ (\mu_Z, \mathcal{C}_Z) & k \text{ even} \end{cases}$$

Proof Technique: Recursive Error Decomposition

Where do these operators come from?

$$(\mu_k, \mathcal{C}_k) := \begin{cases} (\mu_X, \mathcal{C}_X) & k \text{ odd} \\ (\mu_Z, \mathcal{C}_Z) & k \text{ even} \end{cases}$$

$$\begin{aligned}
[P_n^{(k)} - P](h) &= [P_n^{(k)} - P](\mathcal{C}_k h) + \overbrace{[P_n^{(k)} - P](\mu_k h)}^{=0} \\
&= [P_n^{(k-1)} - P](\mathcal{C}_k h) + [P_n^{(k)} - P_n^{(k-1)}](\mathcal{C}_k h) \\
&= \underbrace{[P_n^{(0)} - P](\mathcal{C}_1 \ldots \mathcal{C}_k h)}_{\text{First-Order Term}} + \underbrace{\textstyle\sum_{\ell=1}^{k} [P_n^{(\ell)} - P_n^{(\ell-1)}](\mathcal{C}_\ell \ldots \mathcal{C}_k h)}_{\text{Higher-Order Term}} .
\end{aligned}$$

Proof Technique: Recursive Error Decomposition

Where do these operators come from?

$$(\mu_k, \mathcal{C}_k) := \begin{cases} (\mu_X, \mathcal{C}_X) & k \text{ odd} \\ (\mu_Z, \mathcal{C}_Z) & k \text{ even} \end{cases}$$

$$[P_n^{(k)} - P](h) = [P_n^{(k)} - P](\mathcal{C}_k h) + \overbrace{[P_n^{(k)} - P](\mu_k h)}^{=0}$$
$$= [P_n^{(k-1)} - P](\mathcal{C}_k h) + [P_n^{(k)} - P_n^{(k-1)}](\mathcal{C}_k h)$$
$$= \underbrace{[P_n^{(0)} - P](\mathcal{C}_1 \ldots \mathcal{C}_k h)}_{\text{First-Order Term}} + \underbrace{\textstyle\sum_{\ell=1}^{k} [P_n^{(\ell)} - P_n^{(\ell-1)}](\mathcal{C}_\ell \ldots \mathcal{C}_k h)}_{\text{Higher-Order Term}}.$$

Ex: $\mu_X h$ depends only on marginal P_X, for which they both match.

Proof Technique: Recursive Error Decomposition

Where do these operators come from?

$$(\mu_k, \mathcal{C}_k) := \begin{cases} (\mu_X, \mathcal{C}_X) & k \text{ odd} \\ (\mu_Z, \mathcal{C}_Z) & k \text{ even} \end{cases}$$

$$\begin{aligned}[P_n^{(k)} - P](h) &= [P_n^{(k)} - P](\mathcal{C}_k h) + \overbrace{[P_n^{(k)} - P](\mu_k h)}^{=0} \\ &= [P_n^{(k-1)} - P](\mathcal{C}_k h) + [P_n^{(k)} - P_n^{(k-1)}](\mathcal{C}_k h) \\ &= \underbrace{[P_n^{(0)} - P](\mathcal{C}_1 \ldots \mathcal{C}_k h)}_{\text{First-Order Term}} + \underbrace{\textstyle\sum_{\ell=1}^{k} [P_n^{(\ell)} - P_n^{(\ell-1)}](\mathcal{C}_\ell \ldots \mathcal{C}_k h)}_{\text{Higher-Order Term}} .\end{aligned}$$

Proof Technique: Recursive Error Decomposition

Where do these operators come from?

$$(\mu_k, \mathcal{C}_k) := \begin{cases} (\mu_X, \mathcal{C}_X) & k \text{ odd} \\ (\mu_Z, \mathcal{C}_Z) & k \text{ even} \end{cases}$$

$$\begin{aligned} [P_n^{(k)} - P](h) &= [P_n^{(k)} - P](\mathcal{C}_k h) + \overbrace{[P_n^{(k)} - P](\mu_k h)}^{=0} \\ &= [P_n^{(k-1)} - P](\mathcal{C}_k h) + [P_n^{(k)} - P_n^{(k-1)}](\mathcal{C}_k h) \\ &= \underbrace{[P_n^{(0)} - P](\mathcal{C}_1 \ldots \mathcal{C}_k h)}_{\text{First-Order Term}} + \underbrace{\textstyle\sum_{\ell=1}^k [P_n^{(\ell)} - P_n^{(\ell-1)}](\mathcal{C}_\ell \ldots \mathcal{C}_k h)}_{\text{Higher-Order Term}}. \end{aligned}$$

Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[\left(P_n^{(k)}(h) - P(h)\right)^2\right] = \frac{\mathbb{V}\mathrm{ar}(\mathcal{C}_Z\mathcal{C}_X \dots \mathcal{C}_Z\mathcal{C}_X h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$

$$\begin{aligned}
[P_n^{(k)} - P](h) &= [P_n^{(k)} - P](\mathcal{C}_k h) + \overbrace{[P_n^{(k)} - P](\mu_k h)}^{=0} \\
&= [P_n^{(k-1)} - P](\mathcal{C}_k h) + [P_n^{(k)} - P_n^{(k-1)}](\mathcal{C}_k h) \\
&= \underbrace{[P_n^{(0)} - P](\mathcal{C}_1 \dots \mathcal{C}_k h)}_{\text{First-Order Term}} + \underbrace{\textstyle\sum_{\ell=1}^{k} [P_n^{(\ell)} - P_n^{(\ell-1)}](\mathcal{C}_\ell \dots \mathcal{C}_k h)}_{\text{Higher-Order Term}} .
\end{aligned}$$

Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[\left(P_n^{(k)}(h) - P(h)\right)^2\right] = \boxed{\frac{\mathbb{V}\mathrm{ar}(\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X h)}{n}} + \boxed{\tilde{O}\left(\frac{k^6}{n^{3/2}}\right)}$$

$$\begin{aligned}
[P_n^{(k)} - P](h) &= [P_n^{(k)} - P](\mathcal{C}_k h) + \overbrace{[P_n^{(k)} - P](\mu_k h)}^{=0} \\
&= [P_n^{(k-1)} - P](\mathcal{C}_k h) + [P_n^{(k)} - P_n^{(k-1)}](\mathcal{C}_k h) \\
&= \boxed{\underbrace{[P_n^{(0)} - P](\mathcal{C}_1 \ldots \mathcal{C}_k h)}_{\text{First-Order Term}}} + \boxed{\underbrace{\textstyle\sum_{\ell=1}^{k} [P_n^{(\ell)} - P_n^{(\ell-1)}](\mathcal{C}_\ell \ldots \mathcal{C}_k h)}_{\text{Higher-Order Term}}} .
\end{aligned}$$

Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[\left(P_n^{(k)}(h) - P(h)\right)^2\right] = \frac{\mathbb{V}\mathrm{ar}(\overbrace{\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X}^{k \to \infty\ ?} h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$

Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[\left(P_n^{(k)}(h) - P(h)\right)^2\right] = \frac{\mathbb{V}\mathrm{ar}(\overbrace{\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X}^{k \to \infty\,?} h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$

Note that μ_X and μ_Z are adjoint, meaning they share a *singular value decomposition.*

$$\langle f, \mu_X g\rangle_{\mathbf{L}^2(P)} = \mathbb{E}_P\left[f(X)\mathbb{E}_P\left[g(Z)|X\right]\right] = \mathbb{E}_P\left[f(X)g(Z)\right] = \mathbb{E}_P\left[\mathbb{E}_P\left[f(X)|Z\right]g(Z)\right] = \langle \mu_Z f, g\rangle_{\mathbf{L}^2(P)}$$

Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[\left(P_n^{(k)}(h) - P(h)\right)^2\right] = \frac{\mathbb{V}\mathrm{ar}(\overbrace{\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X}^{k \to \infty\ ?} h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$

Note that μ_X and μ_Z are adjoint, meaning they share a *singular value decomposition.*

$$\langle f, \mu_X g\rangle_{\mathbf{L}^2(P)} = \mathbb{E}_P\left[f(X)\mathbb{E}_P\left[g(Z)|X\right]\right] = \mathbb{E}_P\left[f(X)g(Z)\right] = \mathbb{E}_P\left[\mathbb{E}_P\left[f(X)|Z\right]g(Z)\right] = \langle \mu_Z f, g\rangle_{\mathbf{L}^2(P)}$$

Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[\left(P_n^{(k)}(h) - P(h)\right)^2\right] = \frac{\mathbb{V}\mathrm{ar}(\overbrace{\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X}^{k \to \infty\,?} h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$

Note that μ_X and μ_Z are adjoint, meaning they share a *singular value decomposition.*

$$\langle f, \mu_X g\rangle_{\mathbf{L}^2(P)} = \mathbb{E}_P\left[f(X)\mathbb{E}_P\left[g(Z)|X\right]\right] = \mathbb{E}_P\left[f(X)g(Z)\right] = \mathbb{E}_P\left[\mathbb{E}_P\left[f(X)|Z\right]g(Z)\right] = \langle \mu_Z f, g\rangle_{\mathbf{L}^2(P)}$$

Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[\left(P_n^{(k)}(h) - P(h)\right)^2\right] = \frac{\mathbb{V}\mathrm{ar}(\overbrace{\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X}^{k \to \infty\ ?} h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$

Note that μ_X and μ_Z are adjoint, meaning they share a *singular value decomposition.*

$$\langle f, \mu_X g\rangle_{\mathbf{L}^2(P)} = \mathbb{E}_P\left[f(X)\mathbb{E}_P\left[g(Z)|X\right]\right] = \mathbb{E}_P\left[f(X)g(Z)\right] = \mathbb{E}_P\left[\mathbb{E}_P\left[f(X)|Z\right]g(Z)\right] = \langle \mu_Z f, g\rangle_{\mathbf{L}^2(P)}$$

Basis of $\mathbf{L}^2(P_X)$: $\alpha_1, \alpha_2, \ldots$

$$\mu_X \beta_i = s_i \alpha_i$$

Basis of $\mathbf{L}^2(P_Z)$: $\beta_1, \beta_2, \ldots$

$$\mu_Z \alpha_i = s_i \beta_i$$

Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[\left(P_n^{(k)}(h) - P(h)\right)^2\right] = \frac{\mathbb{V}\mathrm{ar}(\overbrace{\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X}^{k \to \infty\ ?} h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$

Note that μ_X and μ_Z are adjoint, meaning they share a *singular value decomposition.*

$$\langle f, \mu_X g \rangle_{\mathbf{L}^2(P)} = \mathbb{E}_P\left[f(X)\mathbb{E}_P\left[g(Z)|X\right]\right] = \mathbb{E}_P\left[f(X)g(Z)\right] = \mathbb{E}_P\left[\mathbb{E}_P\left[f(X)|Z\right]g(Z)\right] = \langle \mu_Z f, g \rangle_{\mathbf{L}^2(P)}$$

Basis of $\mathbf{L}^2(P_X)$: $\alpha_1, \alpha_2, \ldots$

$$\mu_X \beta_i = s_i \alpha_i$$

Basis of $\mathbf{L}^2(P_Z)$: $\beta_1, \beta_2, \ldots$

$$\mu_Z \alpha_i = s_i \beta_i$$

Singular values = **canonical correlations**.

Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[(P_n^{(k)}(h) - P(h))^2\right] = \frac{\mathbb{V}\mathrm{ar}(\overbrace{\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X}^{k \to \infty\ ?} h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$

Note that μ_X and μ_Z are adjoint, meaning they share a *singular value decomposition*.

$$\langle f, \mu_X g\rangle_{\mathbf{L}^2(P)} = \mathbb{E}_P\left[f(X)\mathbb{E}_P\left[g(Z)|X\right]\right] = \mathbb{E}_P\left[f(X)g(Z)\right] = \mathbb{E}_P\left[\mathbb{E}_P\left[f(X)|Z\right]g(Z)\right] = \langle \mu_Z f, g\rangle_{\mathbf{L}^2(P)}$$

Basis of $\mathbf{L}^2(P_X)$: $\alpha_1, \alpha_2, \ldots$

$$\mu_X\beta_i = s_i\alpha_i$$

Basis of $\mathbf{L}^2(P_Z)$: $\beta_1, \beta_2, \ldots$

$$\mu_Z\alpha_i = s_i\beta_i$$

The sequence of orthogonal complements exhibits a pattern.

$$
\begin{aligned}
\mathcal{C}_X &= I - \mu_X \\
\mathcal{C}_Z\mathcal{C}_X &= I - \mu_X - \mu_Z + \mu_Z\mu_X \\
\mathcal{C}_X\mathcal{C}_Z\mathcal{C}_X &= I - \mu_X - \mu_Z + \mu_Z\mu_X + \mu_X\mu_Z - \mu_X\mu_Z\mu_X,
\end{aligned}
$$

Note that μ_X and μ_Z are adjoint, meaning they share a *singular value decomposition.*

$$\langle f, \mu_X g\rangle_{\mathbf{L}^2(P)} = \mathbb{E}_P\left[f(X)\mathbb{E}_P\left[g(Z)|X\right]\right] = \mathbb{E}_P\left[f(X)g(Z)\right] = \mathbb{E}_P\left[\mathbb{E}_P\left[f(X)|Z\right]g(Z)\right] = \langle \mu_Z f, g\rangle_{\mathbf{L}^2(P)}$$

Basis of $\mathbf{L}^2(P_X)$: $\alpha_1, \alpha_2, \ldots$

$$\mu_X\beta_i = s_i\alpha_i$$

Basis of $\mathbf{L}^2(P_Z)$: $\beta_1, \beta_2, \ldots$

$$\mu_Z\alpha_i = s_i\beta_i$$

The sequence of orthogonal complements exhibits a pattern.

$$\begin{aligned} \mathcal{C}_X &= I - \mu_X \\ \mathcal{C}_Z\mathcal{C}_X &= I - \mu_X - \mu_Z + \mu_Z\mu_X \\ \mathcal{C}_X\mathcal{C}_Z\mathcal{C}_X &= I - \mu_X - \mu_Z + \mu_Z\mu_X + \mu_X\mu_Z - \mu_X\mu_Z\mu_X, \end{aligned} \quad \longmapsto \quad \boxed{\begin{aligned} \mathcal{C}_\ell \ldots \mathcal{C}_k = I - \sum_{\tau=0}^{(k-\ell-1)/2} (\mu_X\mu_Z)^\tau \mu_X - \sum_{\tau=0}^{(k-\ell-1)/2} (\mu_Z\mu_X)^\tau \mu_Z \\ + \sum_{\tau=1}^{(k-\ell)/2} (\mu_X\mu_Z)^\tau + \sum_{\tau=1}^{(k-\ell)/2} (\mu_Z\mu_X)^\tau + (-1)^{k-\ell+1}\mu_\ell \ldots \mu_k, \end{aligned}}$$

Note that μ_X and μ_Z are adjoint, meaning they share a *singular value decomposition.*

$$\langle f, \mu_X g\rangle_{\mathbf{L}^2(P)} = \mathbb{E}_P\left[f(X)\mathbb{E}_P\left[g(Z)|X\right]\right] = \mathbb{E}_P\left[f(X)g(Z)\right] = \mathbb{E}_P\left[\mathbb{E}_P\left[f(X)|Z\right]g(Z)\right] = \langle \mu_Z f, g\rangle_{\mathbf{L}^2(P)}$$

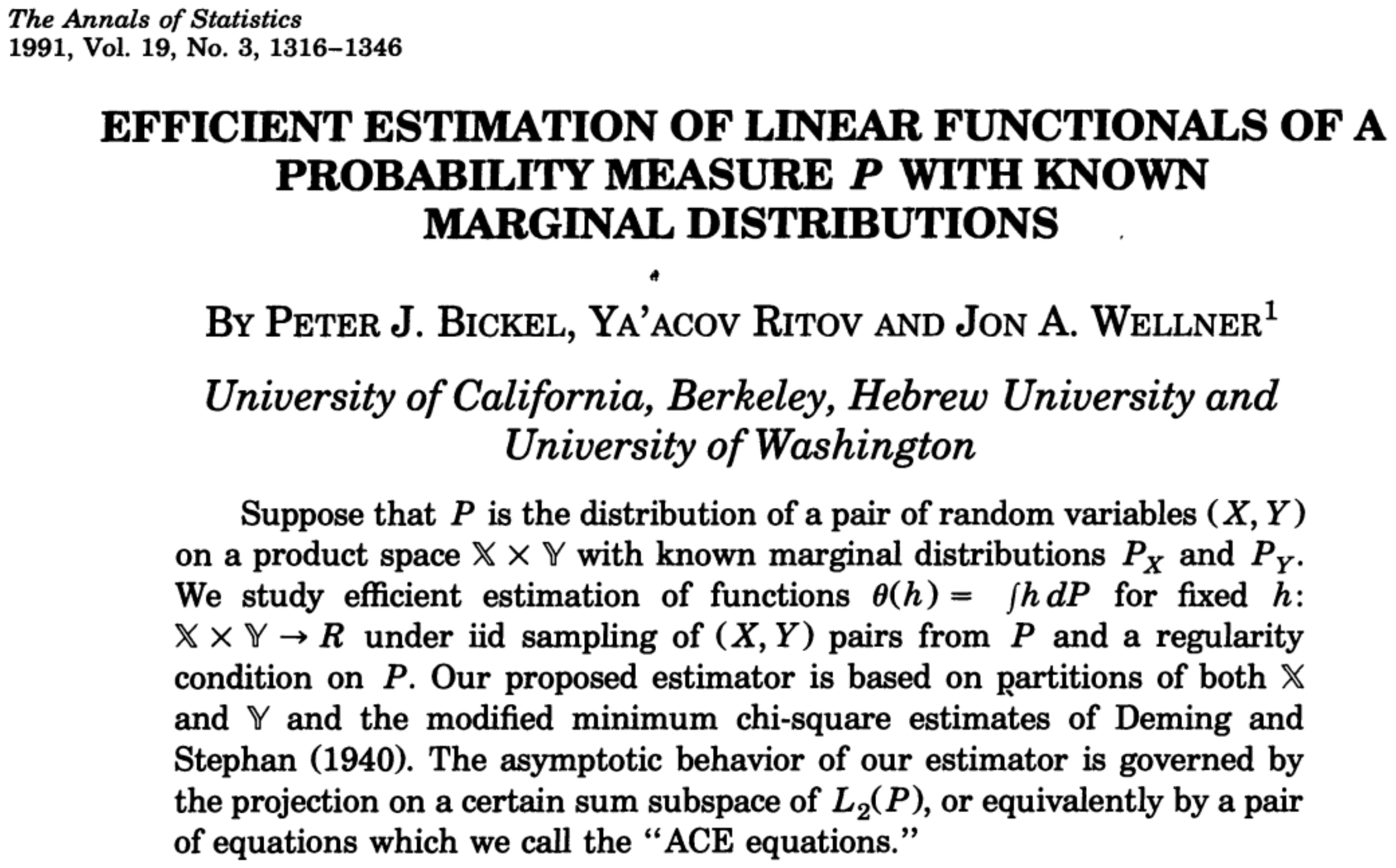
Basis of $\mathbf{L}^2(P_X)$: $\alpha_1, \alpha_2, \ldots$

$$\mu_X \beta_i = s_i \alpha_i$$

Basis of $\mathbf{L}^2(P_Z)$: $\beta_1, \beta_2, \ldots$

$$\mu_Z \alpha_i = s_i \beta_i$$

Quantifying this variance reduction is a classical problem in mathematical statistics, particularly efficiency theory.



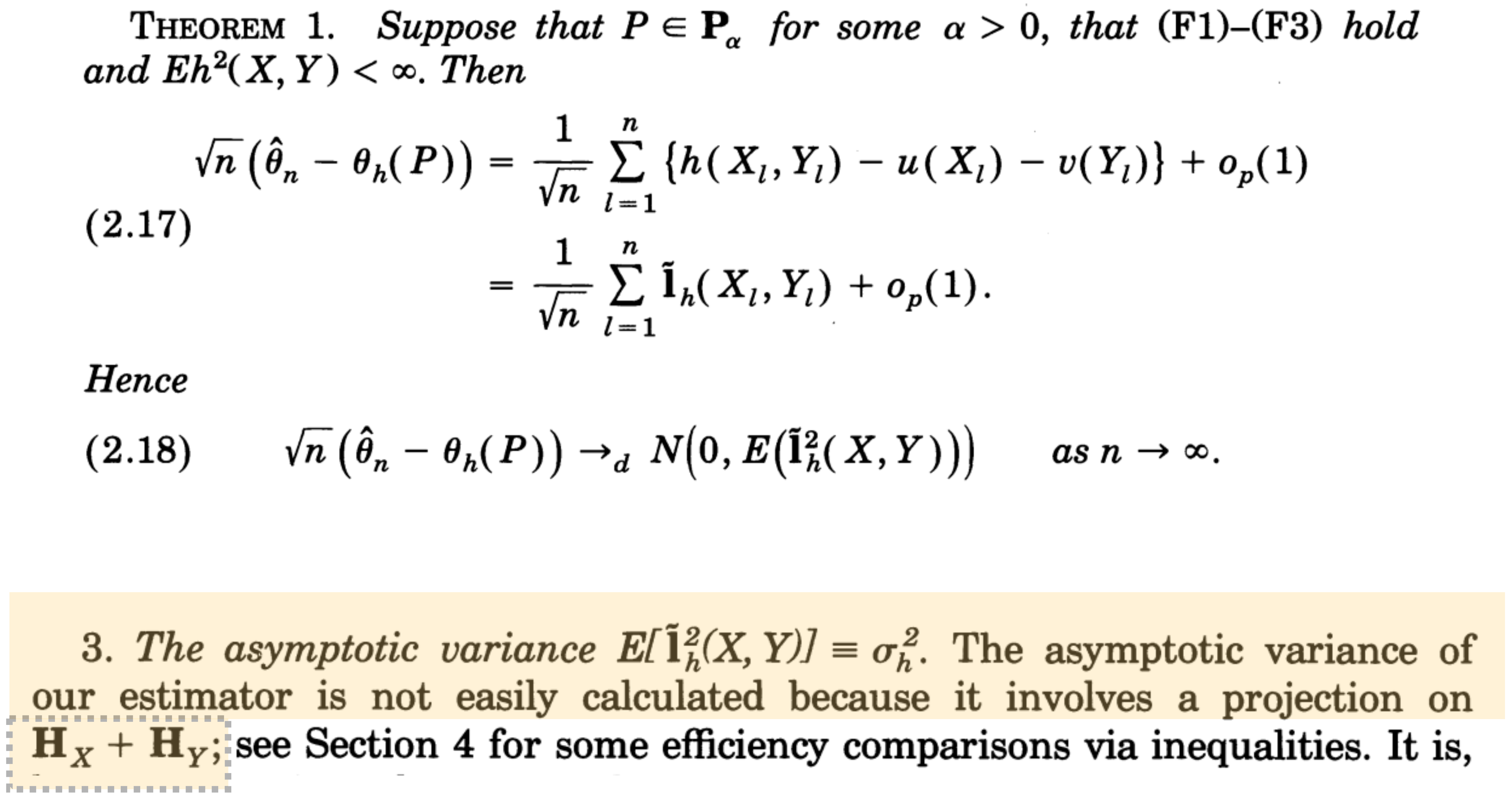
The Annals of Statistics
1991, Vol. 19, No. 3, 1316–1346

EFFICIENT ESTIMATION OF LINEAR FUNCTIONALS OF A PROBABILITY MEASURE P WITH KNOWN MARGINAL DISTRIBUTIONS

BY PETER J. BICKEL, YA'ACOV RITOV AND JON A. WELLNER[1]

University of California, Berkeley, Hebrew University and University of Washington

Suppose that P is the distribution of a pair of random variables (X, Y) on a product space $\mathbb{X} \times \mathbb{Y}$ with known marginal distributions P_X and P_Y. We study efficient estimation of functions $\theta(h) = \int h\,dP$ for fixed h: $\mathbb{X} \times \mathbb{Y} \to R$ under iid sampling of (X, Y) pairs from P and a regularity condition on P. Our proposed estimator is based on partitions of both $\mathbb{X}$ and $\mathbb{Y}$ and the modified minimum chi-square estimates of Deming and Stephan (1940). The asymptotic behavior of our estimator is governed by the projection on a certain sum subspace of $L_2(P)$, or equivalently by a pair of equations which we call the "ACE equations."



THEOREM 1. *Suppose that* $P \in \mathbf{P}_\alpha$ *for some* $\alpha > 0$, *that* (F1)–(F3) *hold and* $Eh^2(X, Y) < \infty$. *Then*

$$(2.17)\quad \begin{aligned} \sqrt{n}\left(\hat{\theta}_n - \theta_h(P)\right) &= \frac{1}{\sqrt{n}} \sum_{l=1}^{n} \{h(X_l, Y_l) - u(X_l) - v(Y_l)\} + o_p(1) \\ &= \frac{1}{\sqrt{n}} \sum_{l=1}^{n} \tilde{\mathbf{l}}_h(X_l, Y_l) + o_p(1). \end{aligned}$$

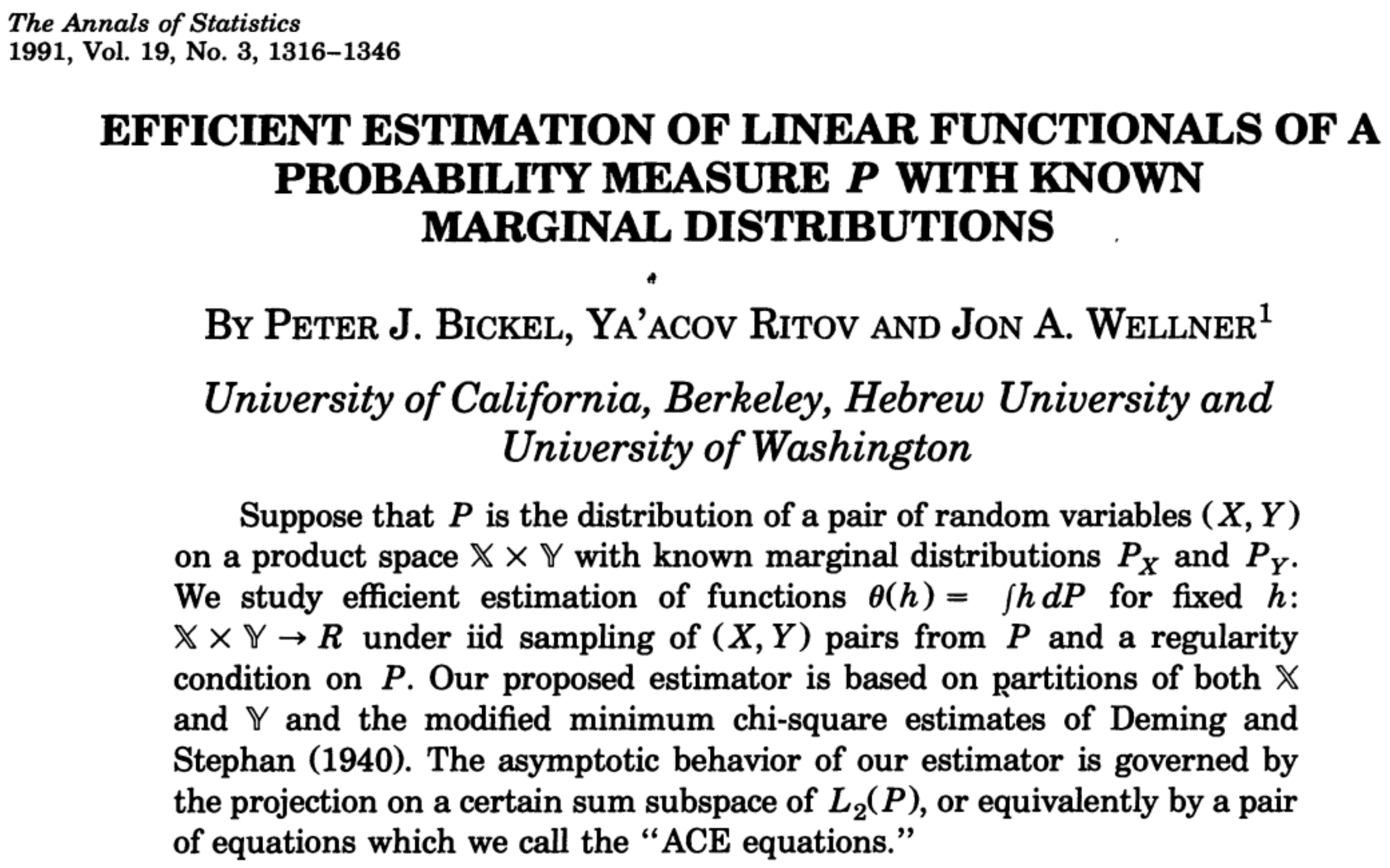
Hence

$$(2.18)\quad \sqrt{n}\left(\hat{\theta}_n - \theta_h(P)\right) \to_d N\left(0, E\left(\tilde{\mathbf{l}}_h^2(X, Y)\right)\right) \qquad \text{as } n \to \infty.$$

3. *The asymptotic variance* $E[\tilde{\mathbf{l}}_h^2(X, Y)] \equiv \sigma_h^2$. The asymptotic variance of our estimator is not easily calculated because it involves a projection on $\boxed{\mathbf{H}_X + \mathbf{H}_Y;}$ see Section 4 for some efficiency comparisons via inequalities. It is,

$\mathbf{L}^2(P_X)^\perp \cap \mathbf{L}^2(P_Z)^\perp$

Quantifying this variance reduction is a classical problem in mathematical statistics, particularly efficiency theory.



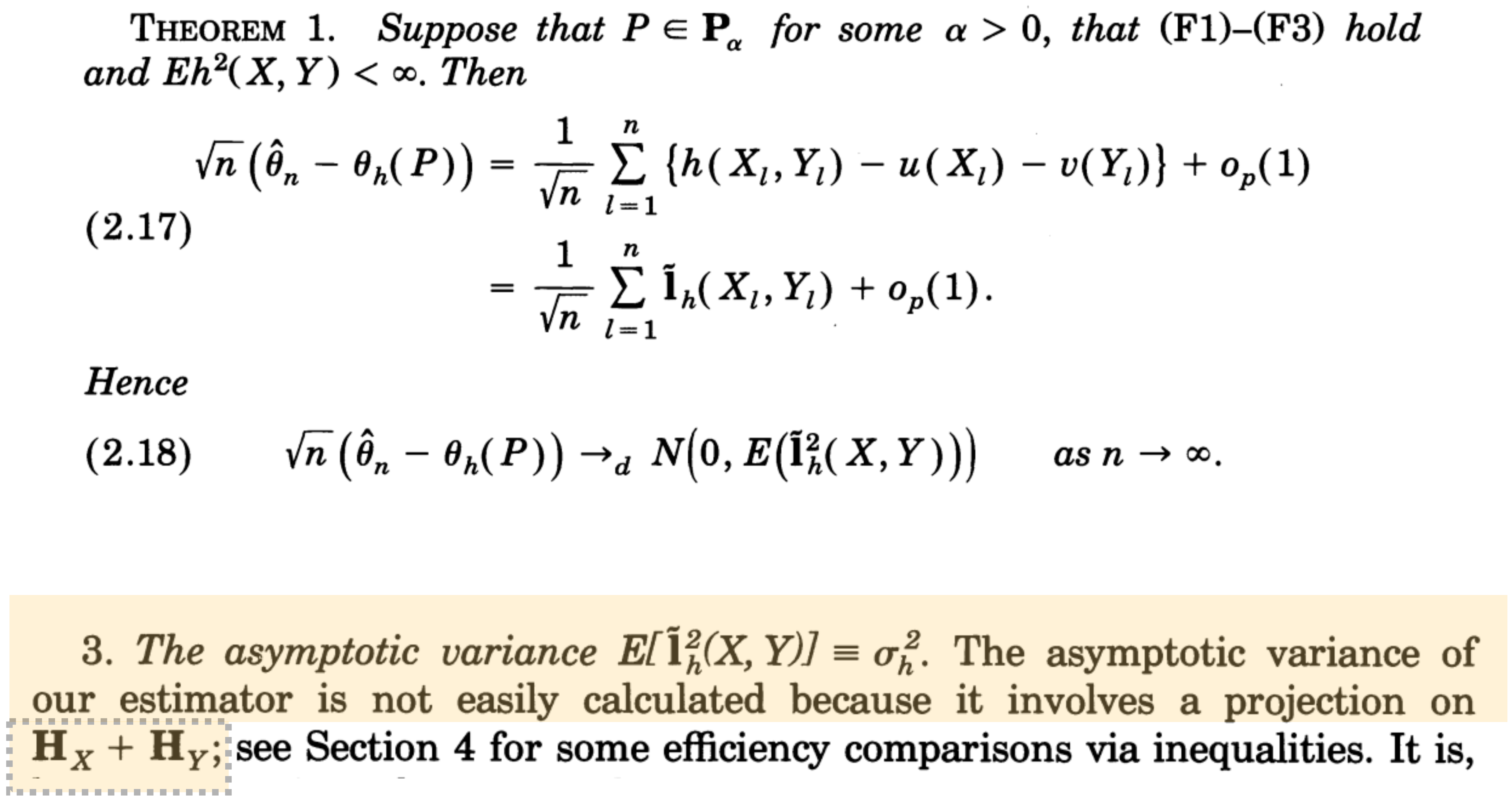
The Annals of Statistics
1991, Vol. 19, No. 3, 1316–1346

EFFICIENT ESTIMATION OF LINEAR FUNCTIONALS OF A PROBABILITY MEASURE P WITH KNOWN MARGINAL DISTRIBUTIONS

BY PETER J. BICKEL, YA'ACOV RITOV AND JON A. WELLNER[1]

University of California, Berkeley, Hebrew University and University of Washington

Suppose that P is the distribution of a pair of random variables (X, Y) on a product space $\mathbb{X} \times \mathbb{Y}$ with known marginal distributions P_X and P_Y. We study efficient estimation of functions $\theta(h) = \int h\,dP$ for fixed h: $\mathbb{X} \times \mathbb{Y} \to R$ under iid sampling of (X, Y) pairs from P and a regularity condition on P. Our proposed estimator is based on partitions of both $\mathbb{X}$ and $\mathbb{Y}$ and the modified minimum chi-square estimates of Deming and Stephan (1940). The asymptotic behavior of our estimator is governed by the projection on a certain sum subspace of $L_2(P)$, or equivalently by a pair of equations which we call the "ACE equations."



THEOREM 1. *Suppose that* $P \in \mathbf{P}_\alpha$ *for some* $\alpha > 0$, *that* (F1)–(F3) *hold and* $Eh^2(X, Y) < \infty$. *Then*

$$(2.17)\qquad \begin{aligned}\sqrt{n}\left(\hat{\theta}_n - \theta_h(P)\right) &= \frac{1}{\sqrt{n}} \sum_{l=1}^{n} \{h(X_l, Y_l) - u(X_l) - v(Y_l)\} + o_p(1)\\ &= \frac{1}{\sqrt{n}} \sum_{l=1}^{n} \tilde{\mathbf{I}}_h(X_l, Y_l) + o_p(1).\end{aligned}$$

Hence

$$(2.18)\qquad \sqrt{n}\left(\hat{\theta}_n - \theta_h(P)\right) \to_d N\left(0, E\left(\tilde{\mathbf{I}}_h^2(X, Y)\right)\right) \qquad \text{as } n \to \infty.$$

3. *The asymptotic variance* $E[\tilde{\mathbf{I}}_h^2(X, Y)] \equiv \sigma_h^2$. The asymptotic variance of our estimator is not easily calculated because it involves a projection on $\boxed{\mathbf{H}_X + \mathbf{H}_Y;}$ see Section 4 for some efficiency comparisons via inequalities. It is,

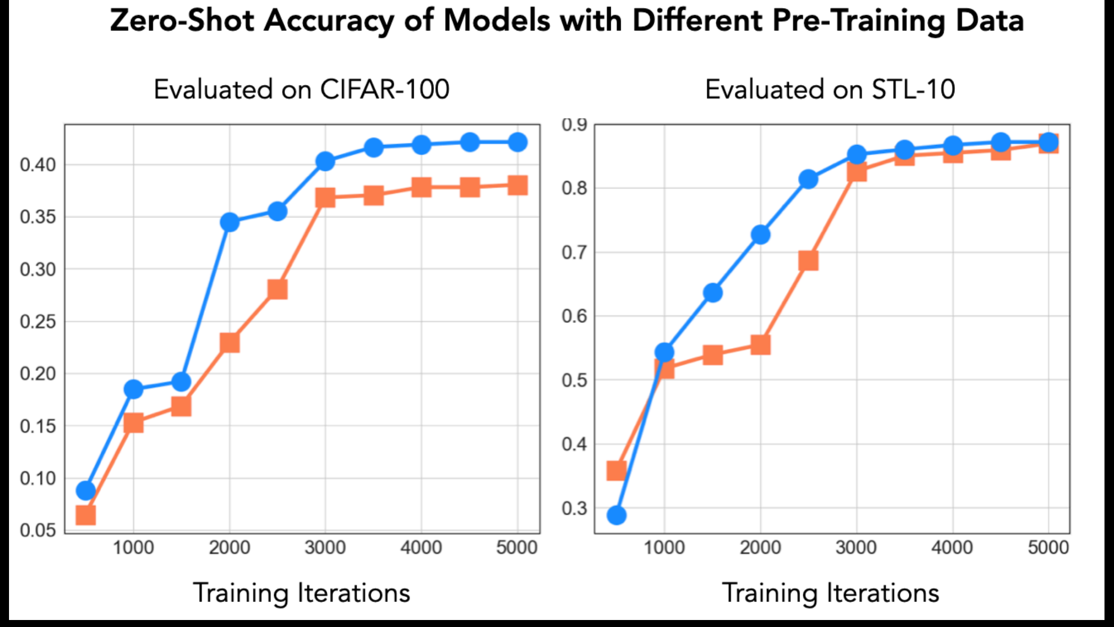
$\mathbf{L}^2(P_X)^\perp \cap \mathbf{L}^2(P_Z)^\perp$

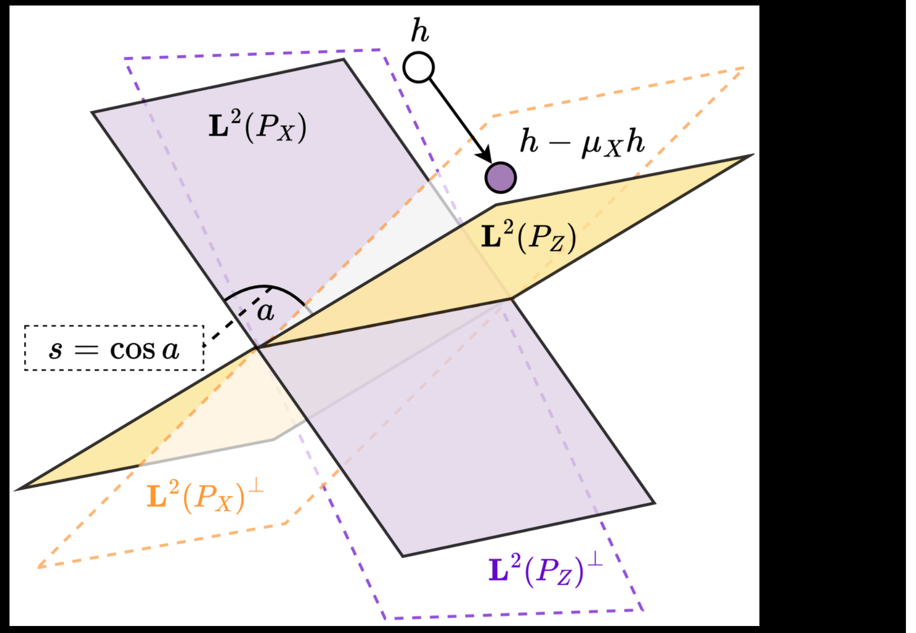
We used a particular optimization algorithm used to **compute** an estimator, in order to analyze it **statistically**. Every iterate of the algorithm has a closed form, but the limit does not.

Contributions. We show that:

The data curation procedure used in CLIP is an instance of balancing at the **pre-training set scale**.

We quantify the theoretical improvement of using such a procedure in terms of variance-reduced estimation of the population loss.



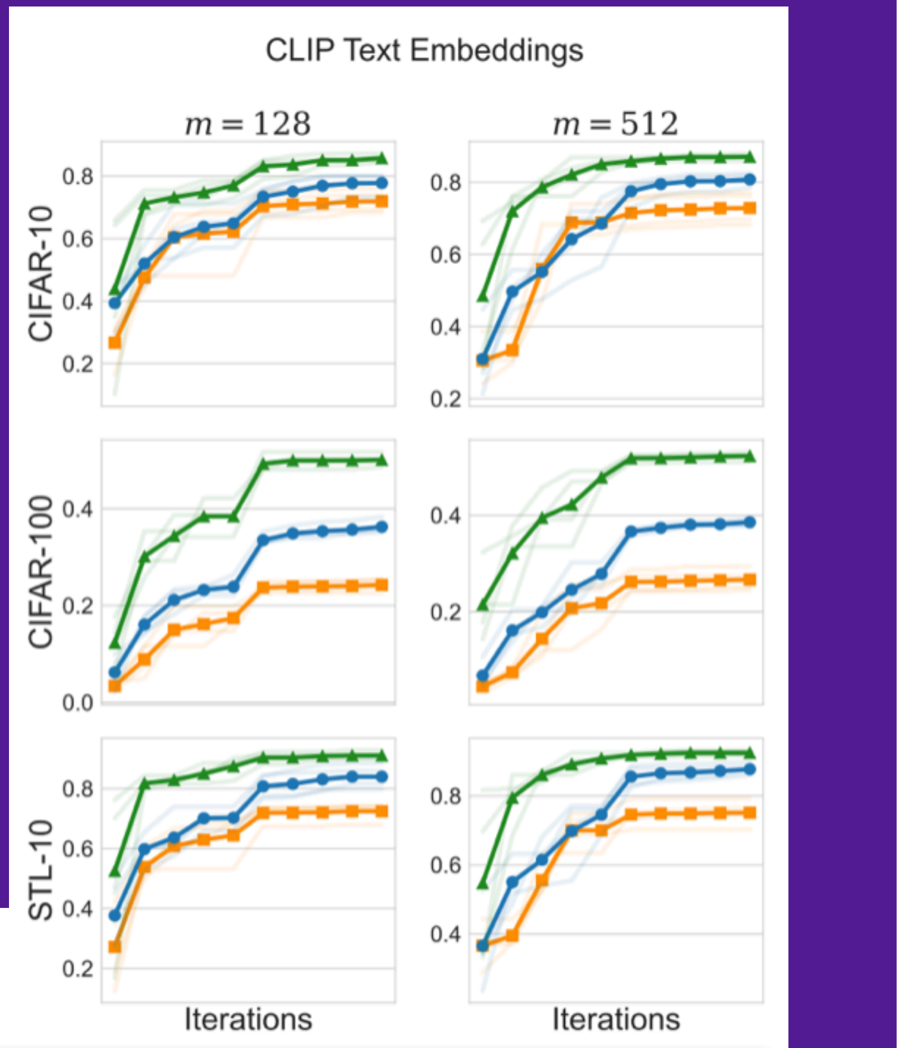


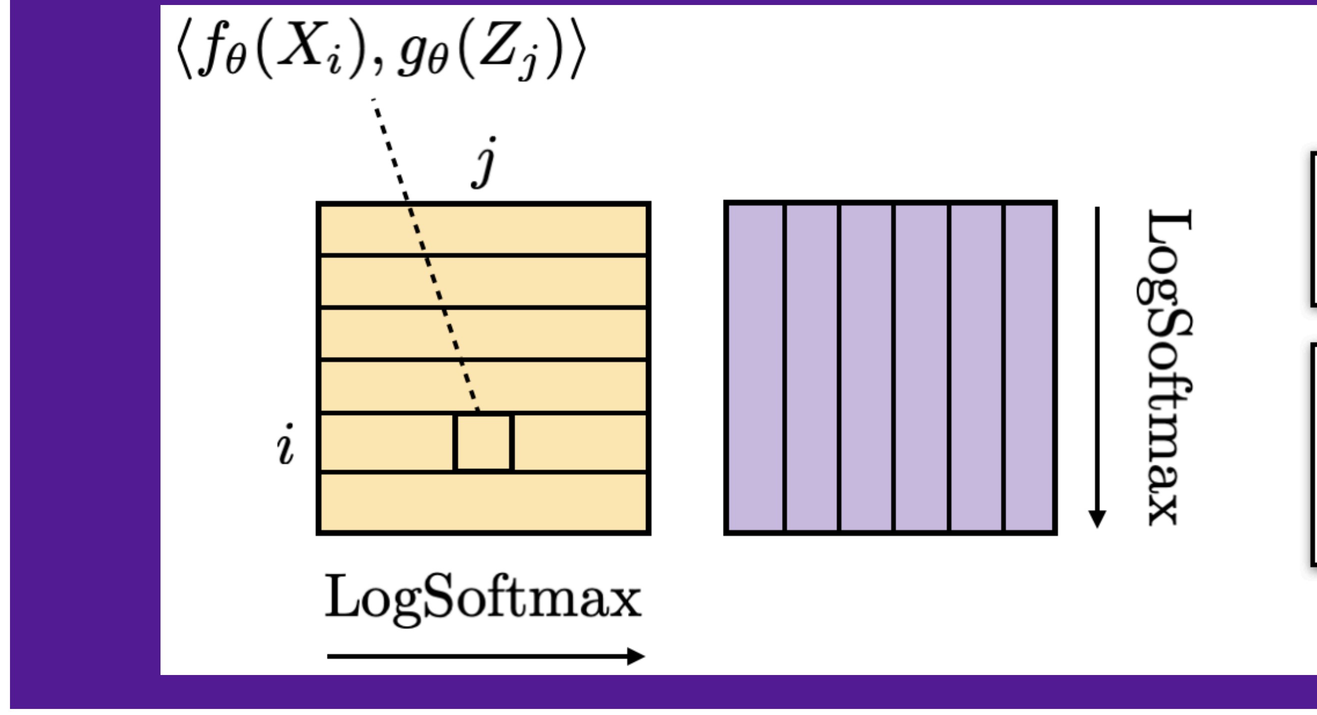
Theorem (Liu, M., Pal, Harchaoui)

$$\mathbb{E}_{P^n}\left[(P_n^{(k)}(h) - P(h))^2\right] = \frac{\mathrm{Var}(\overbrace{\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X}^{k \text{ times}} h)}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right)$$

The CLIP objective computes a functional balanced probability measure at the **mini-batch scale**.

We use this viewpoint to propose an alternative CLIP-like objective that improves zero-shot classification performance empirically.

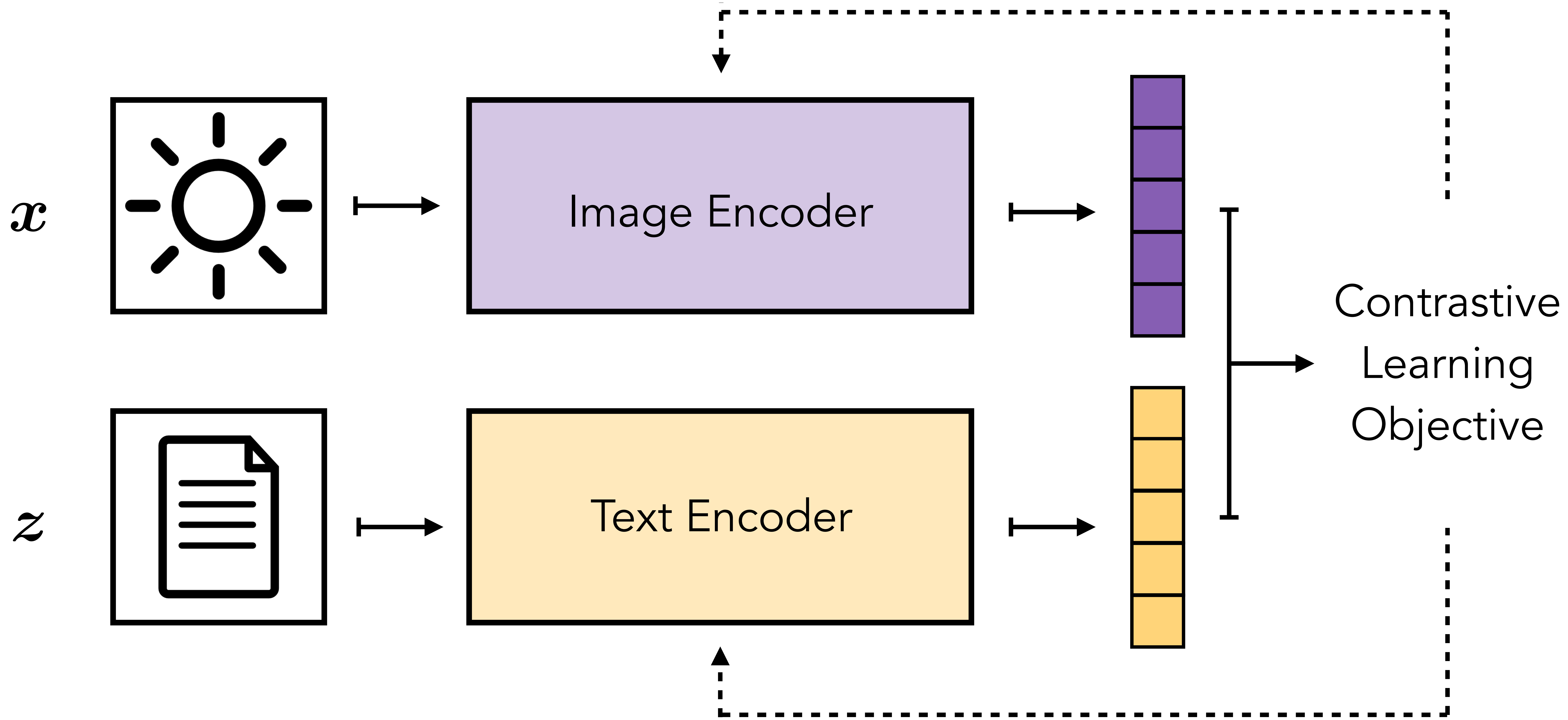




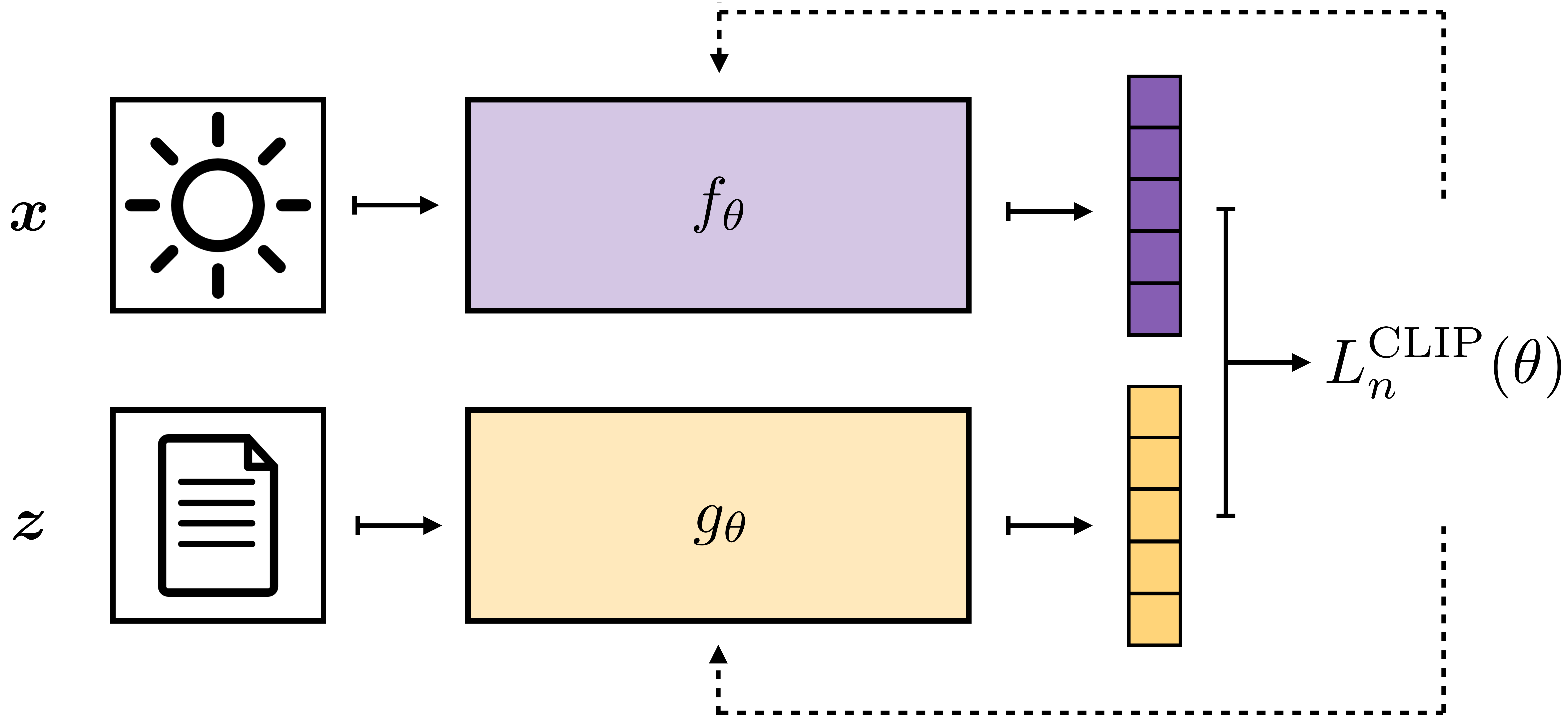
```
def clip_loss(logits):
    cx   = F.log_softmax(logits, dim=1)
    cy   = F.log_softmax(logits, dim=0)
    return -torch.mean(0.5 * torch.diagonal(cx) + 0.5 * torch.diagonal(cy))
```

```
def doubly_centered_loss(logits):
    cx   = F.log_softmax(logits, dim=1)
    cy   = F.log_softmax(logits, dim=0)
    cycx = F.log_softmax(cx, dim=0)
    cxcy = F.log_softmax(cy, dim=1)
    return -torch.mean(0.5 * torch.diagonal(cycx) + 0.5 * torch.diagonal(cxcy))
```

The CLIP objective compute graph contains a *backpropable* balancing step.



The CLIP objective compute graph contains a *backpropable* balancing step.



The CLIP objective compute graph contains a *backpropable* balancing step.

$$L_n^{\mathrm{CLIP}}(\theta) = -\frac{1}{2}\sum_{i=1}^{n}\left[\log \frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_i), g_\theta(Z_j)\rangle}} + \log \frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_j), g_\theta(Z_i)\rangle}}\right]$$

The CLIP objective compute graph contains a *backpropable* balancing step.

$$L_n^{\mathrm{CLIP}}(\theta) = -\frac{1}{2}\sum_{i=1}^{n}\left[\log \frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_i), g_\theta(Z_j)\rangle}} + \log \frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_j), g_\theta(Z_i)\rangle}}\right]$$

$$P_n^{(0)}(\boldsymbol{x}, \boldsymbol{z}) := e^{\langle f_\theta(\boldsymbol{x}), g_\theta(\boldsymbol{z})\rangle}$$

The CLIP objective compute graph contains a *backpropable* balancing step.

$$L_n^{\mathrm{CLIP}}(\theta) = -\frac{1}{2}\sum_{i=1}^{n}\left[\log\frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_i), g_\theta(Z_j)\rangle}} + \log\frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_j), g_\theta(Z_i)\rangle}}\right]$$

$$= -\frac{1}{2}\left[\log\frac{P_n^{(0)}(X_i, Z_i)}{P_{n,X}^{(0)}(X_i)} + \log\frac{P_n^{(0)}(X_i, Z_i)}{P_{n,Z}^{(0)}(Z_i)}\right]$$

$$P_n^{(0)}(\boldsymbol{x}, \boldsymbol{z}) := e^{\langle f_\theta(\boldsymbol{x}), g_\theta(\boldsymbol{z})\rangle}$$

The CLIP objective compute graph contains a *backpropable* balancing step.

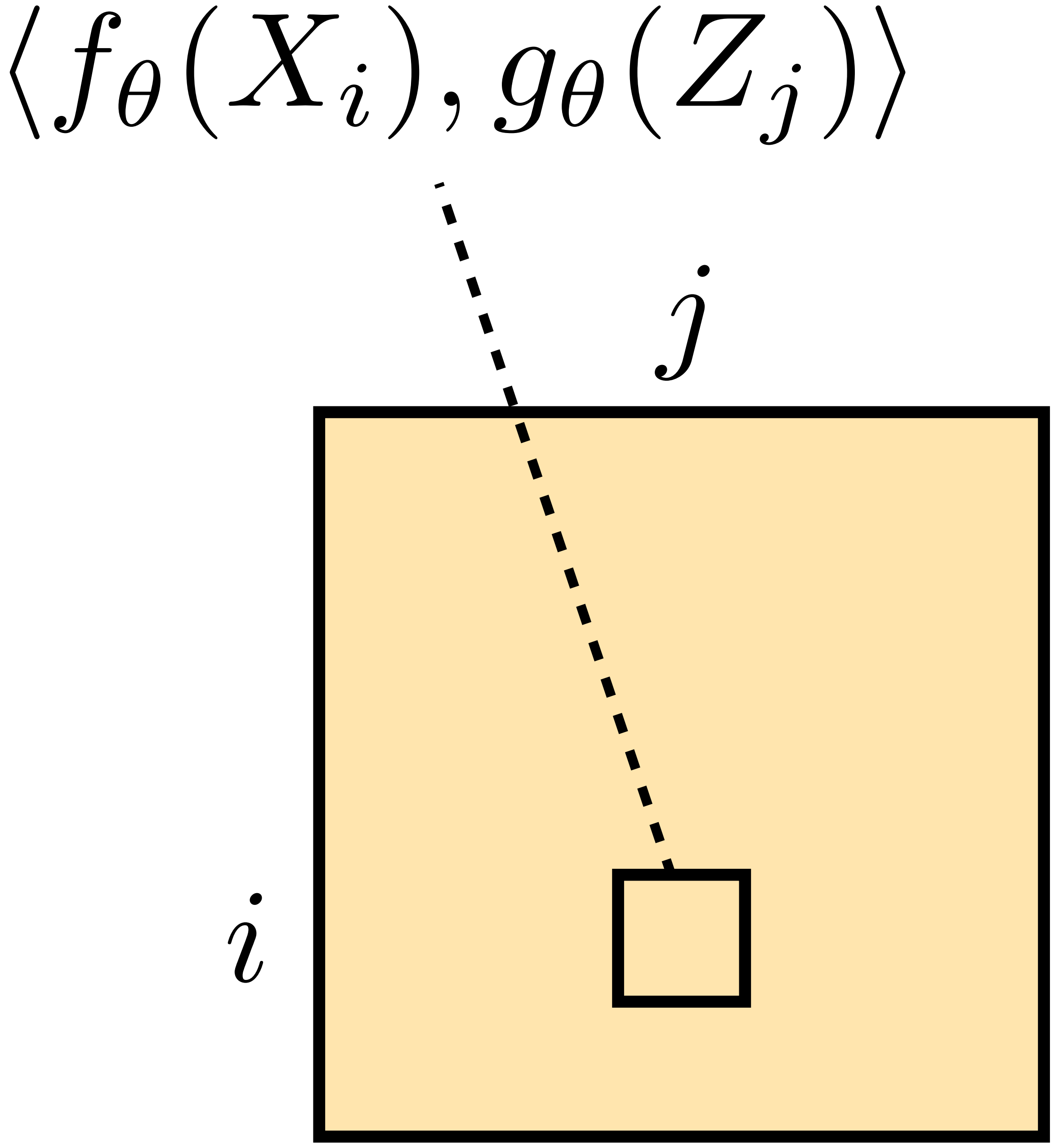
$$
\begin{aligned}
L_n^{\mathrm{CLIP}}(\theta) &= -\frac{1}{2}\sum_{i=1}^{n}\left[\log \frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_i), g_\theta(Z_j)\rangle}} + \log \frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_j), g_\theta(Z_i)\rangle}}\right] \\
&= -\frac{1}{2}\left[\log \frac{P_n^{(0)}(X_i, Z_i)}{P_{n,X}^{(0)}(X_i)} + \log \frac{P_n^{(0)}(X_i, Z_i)}{P_{n,Z}^{(0)}(Z_i)}\right] \\
&= -\frac{1}{2}\left[\log\left(\frac{1/n}{P_{n,X}^{(0)}(X_i)} \cdot P_n^{(0)}(X_i, Z_i)\right) + \log\left(\frac{1/n}{P_{n,Z}^{(0)}(Z_i)} \cdot P_n^{(0)}(X_i, Z_i)\right)\right] - \log n
\end{aligned}
$$

The CLIP objective compute graph contains a *backpropable* balancing step.

$$
\begin{aligned}
L_n^{\mathrm{CLIP}}(\theta) &= -\frac{1}{2}\sum_{i=1}^{n}\left[\log\frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_i), g_\theta(Z_j)\rangle}} + \log\frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_j), g_\theta(Z_i)\rangle}}\right] \\
&= -\frac{1}{2}\left[\log\frac{P_n^{(0)}(X_i, Z_i)}{P_{n,X}^{(0)}(X_i)} + \log\frac{P_n^{(0)}(X_i, Z_i)}{P_{n,Z}^{(0)}(Z_i)}\right] \\
&= -\frac{1}{2}\left[\log\left(\underbrace{\frac{1/n}{P_{n,X}^{(0)}(X_i)} \cdot P_n^{(0)}(X_i, Z_i)}_{\substack{P_n^{(1)}(X_i, Z_i) \\ \text{if balancing } X \text{ first}}}\right) + \log\left(\underbrace{\frac{1/n}{P_{n,Z}^{(0)}(Z_i)} \cdot P_n^{(0)}(X_i, Z_i)}_{\substack{P_n^{(1)}(X_i, Z_i) \\ \text{if balancing } Z \text{ first}}}\right)\right] - \log n
\end{aligned}
$$

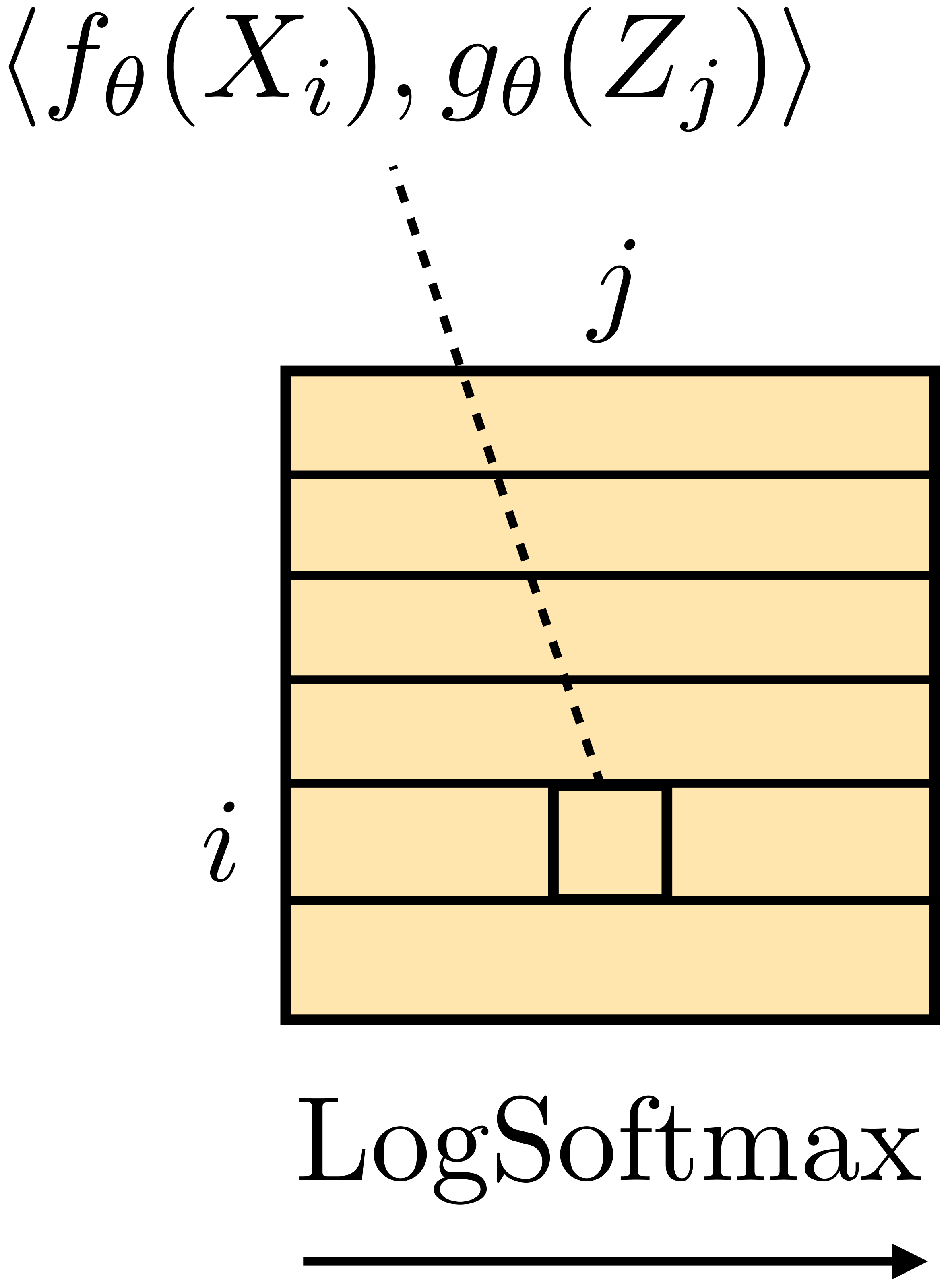
The CLIP objective compute graph contains a *backpropable* balancing step.

$$L_n^{\text{CLIP}}(\theta) = -\frac{1}{2}\sum_{i=1}^{n}\left[\log\frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_i), g_\theta(Z_j)\rangle}} + \log\frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_j), g_\theta(Z_i)\rangle}}\right]$$



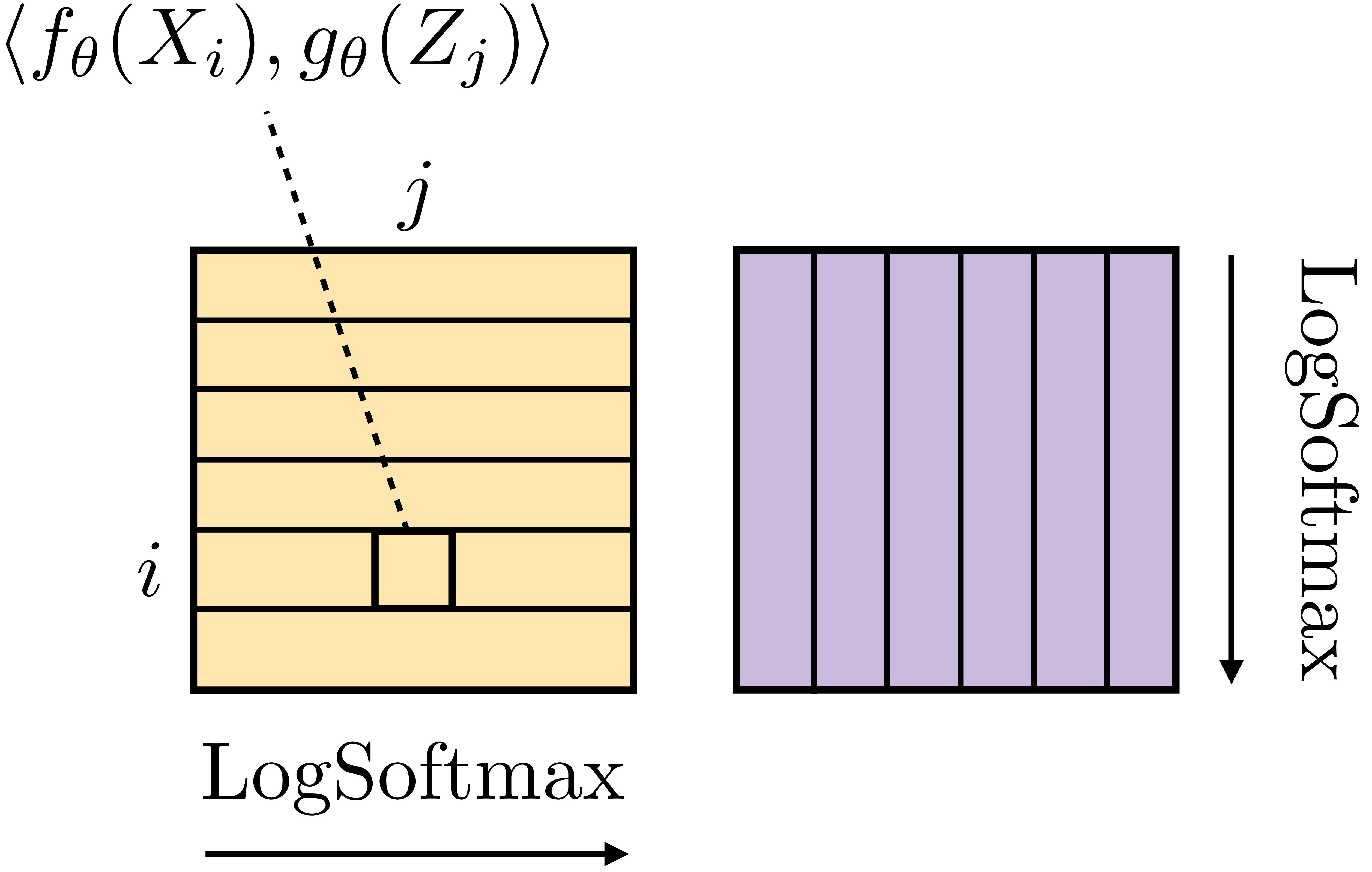
The CLIP objective compute graph contains a *backpropable* balancing step.

$$L_n^{\mathrm{CLIP}}(\theta) = -\frac{1}{2}\sum_{i=1}^{n}\left[\log\frac{e^{\langle f_\theta(X_i),g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_i),g_\theta(Z_j)\rangle}} + \log\frac{e^{\langle f_\theta(X_i),g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_j),g_\theta(Z_i)\rangle}}\right]$$



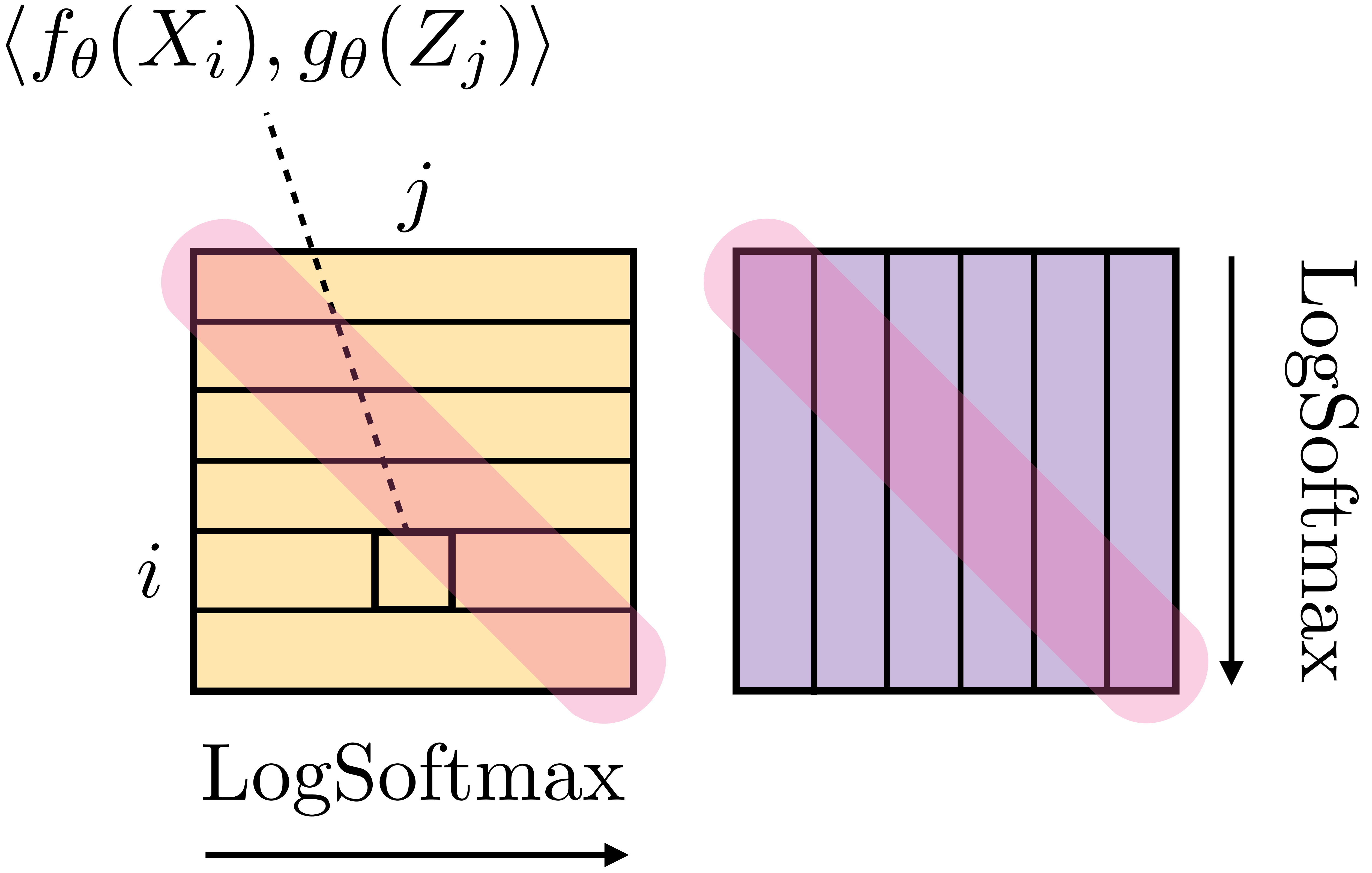
The CLIP objective compute graph contains a *backpropable* balancing step.

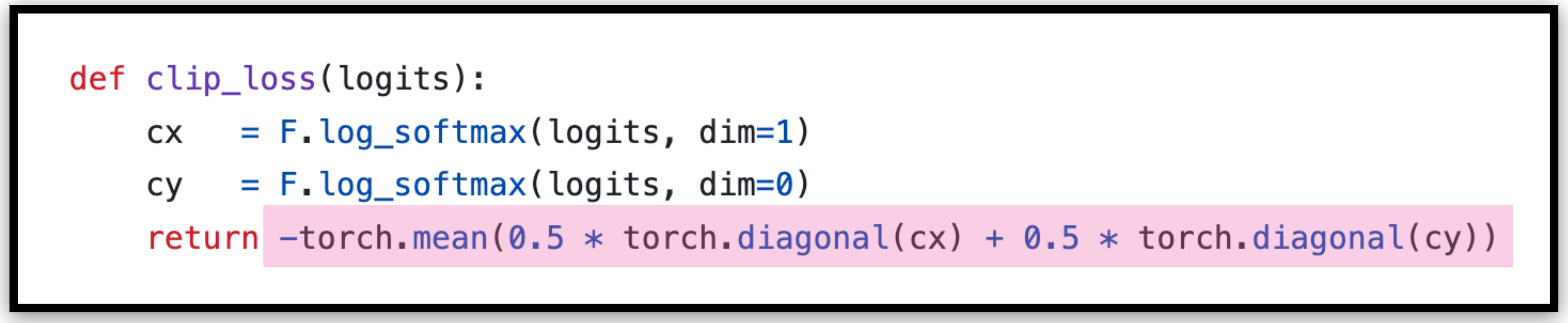
$$L_n^{\mathrm{CLIP}}(\theta) = -\frac{1}{2}\sum_{i=1}^{n}\left[\log \frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_i), g_\theta(Z_j)\rangle}} + \log \frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_j), g_\theta(Z_i)\rangle}}\right]$$



The CLIP objective compute graph contains a *backpropable* balancing step.

$$L_n^{\text{CLIP}}(\theta) = -\frac{1}{2}\sum_{i=1}^{n} \left[\log \frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_i), g_\theta(Z_j)\rangle}} + \log \frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_j), g_\theta(Z_i)\rangle}}\right]$$

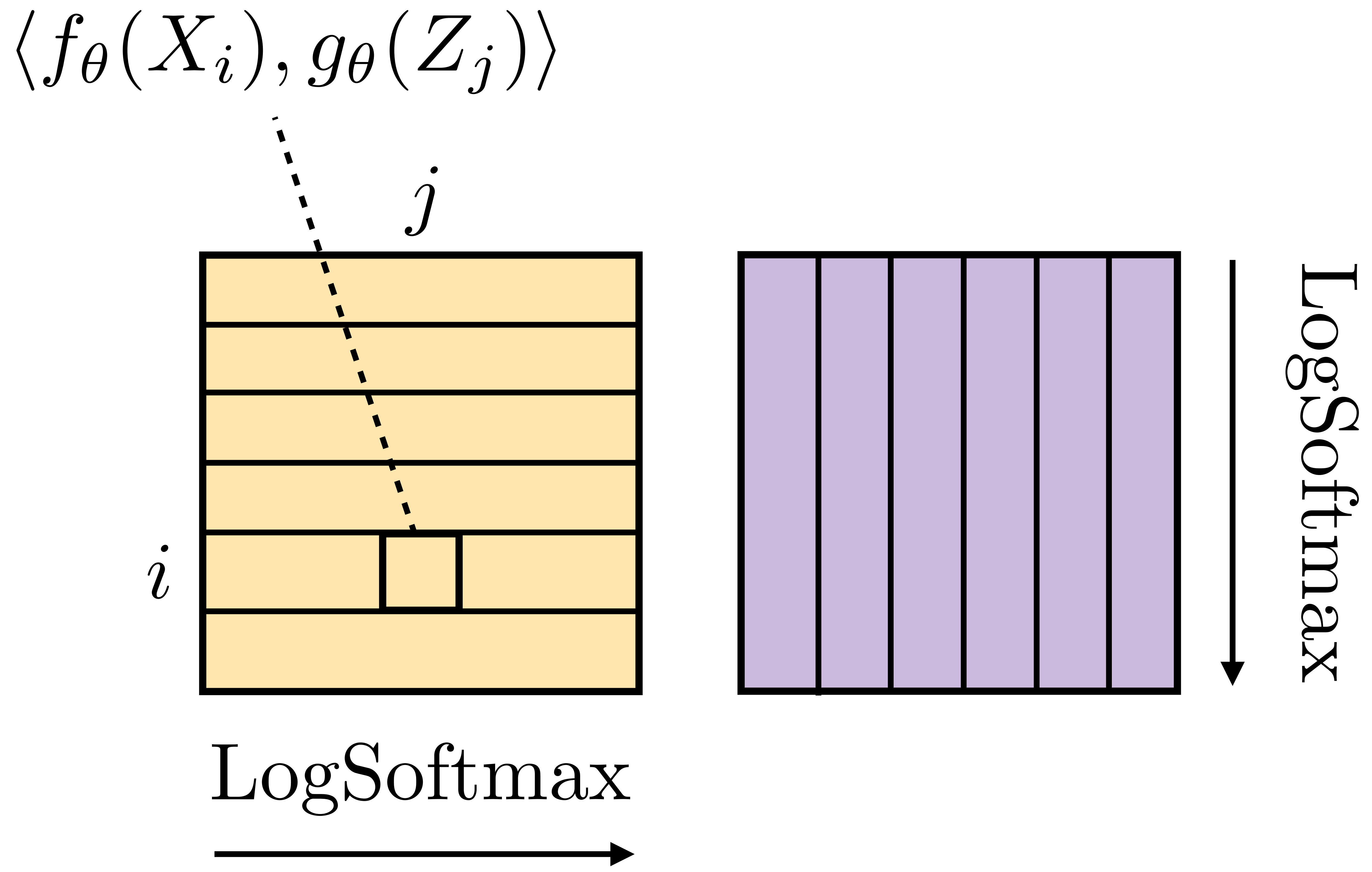


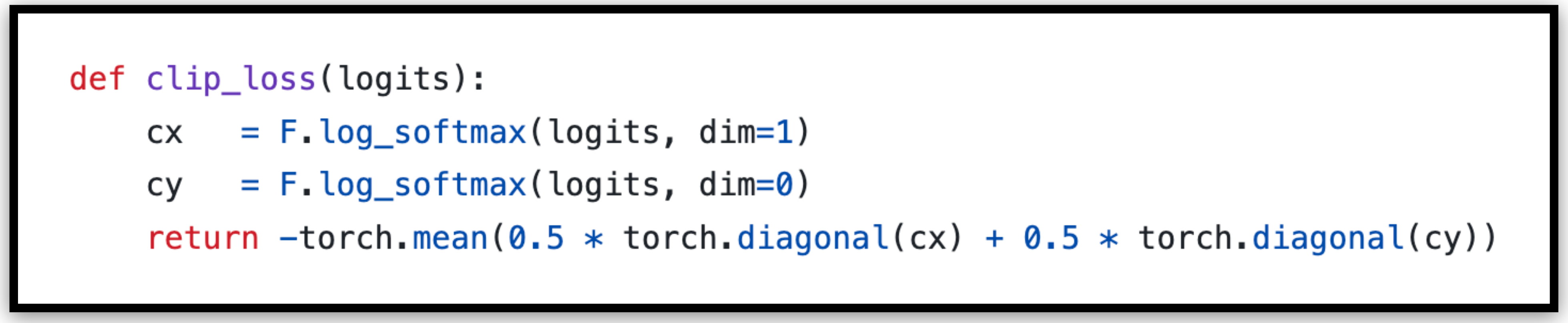


```
def clip_loss(logits):
    cx   = F.log_softmax(logits, dim=1)
    cy   = F.log_softmax(logits, dim=0)
    return -torch.mean(0.5 * torch.diagonal(cx) + 0.5 * torch.diagonal(cy))
```

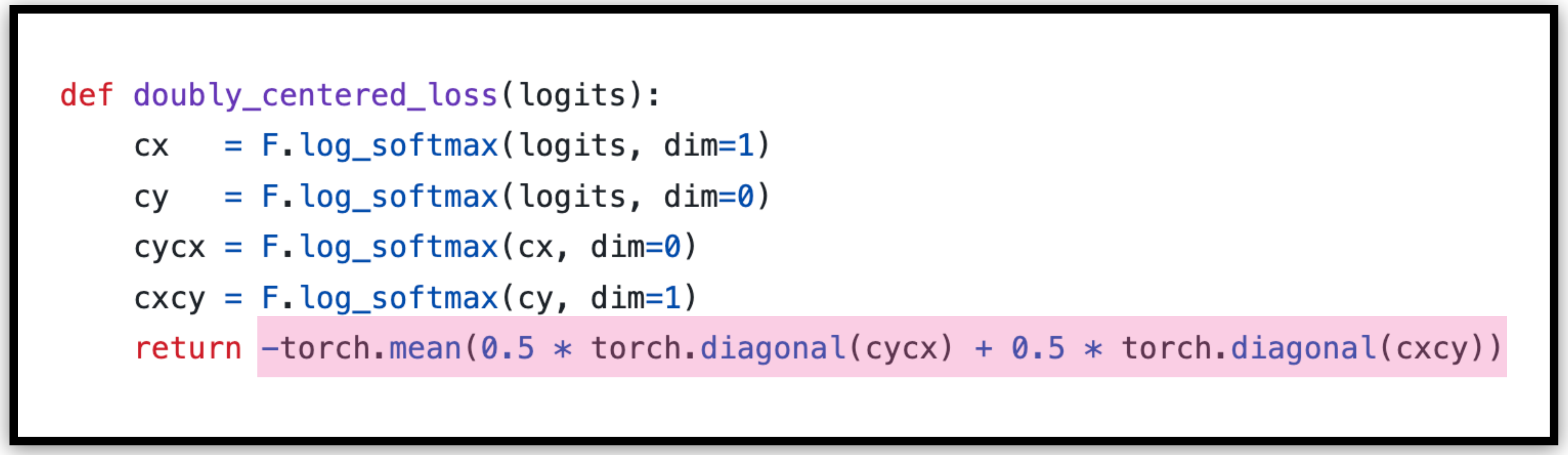
The CLIP objective compute graph contains a *backpropable* balancing step.

$$L_n^{\text{CLIP}}(\theta) = -\frac{1}{2}\sum_{i=1}^{n}\left[\log \frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_i), g_\theta(Z_j)\rangle}} + \log \frac{e^{\langle f_\theta(X_i), g_\theta(Z_i)\rangle}}{\sum_{j=1}^{n} e^{\langle f_\theta(X_j), g_\theta(Z_i)\rangle}}\right]$$





```
def clip_loss(logits):
    cx   = F.log_softmax(logits, dim=1)
    cy   = F.log_softmax(logits, dim=0)
    return -torch.mean(0.5 * torch.diagonal(cx) + 0.5 * torch.diagonal(cy))
```



```
def doubly_centered_loss(logits):
    cx   = F.log_softmax(logits, dim=1)
    cy   = F.log_softmax(logits, dim=0)
    cycx = F.log_softmax(cx, dim=0)
    cxcy = F.log_softmax(cy, dim=1)
    return -torch.mean(0.5 * torch.diagonal(cycx) + 0.5 * torch.diagonal(cxcy))
```

Increasing the number of iterations results in zero-shot accuracy gains!

Increasing the number of iterations results in zero-shot accuracy gains!

CLIP Text Embeddings

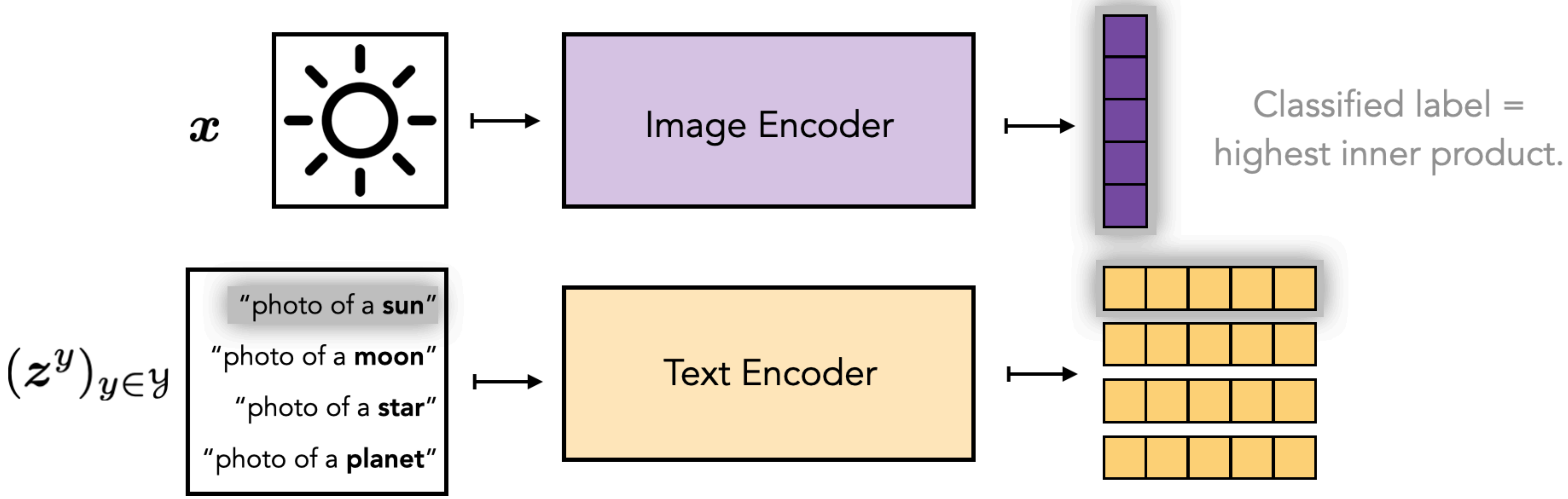
BERT Text Embeddings

GPT-2 Text Embeddings

CIFAR-10

CIFAR-100

STL-10



No Balancing — CLIP Balancing ($k = 1$) — Multi-CLIP ($k = 2$)

Increasing the number of iterations results in zero-shot accuracy gains!

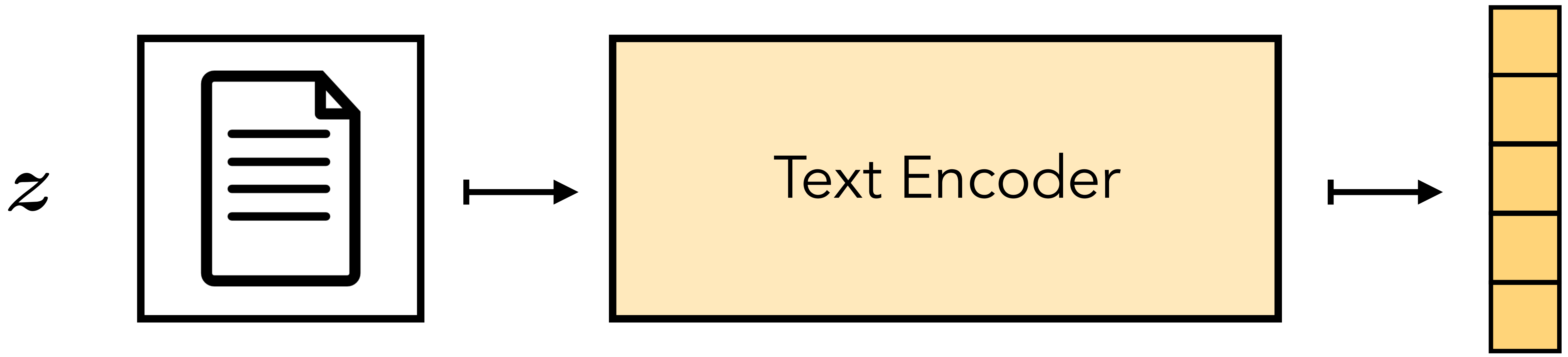
CLIP Text Embeddings

BERT Text Embeddings

GPT-2 Text Embeddings

CIFAR-10

CIFAR-100



STL-10

No Balancing CLIP Balancing ($k = 1$) Multi-CLIP ($k = 2$)

Increasing the number of iterations results in zero-shot accuracy gains!

CLIP Text Embeddings

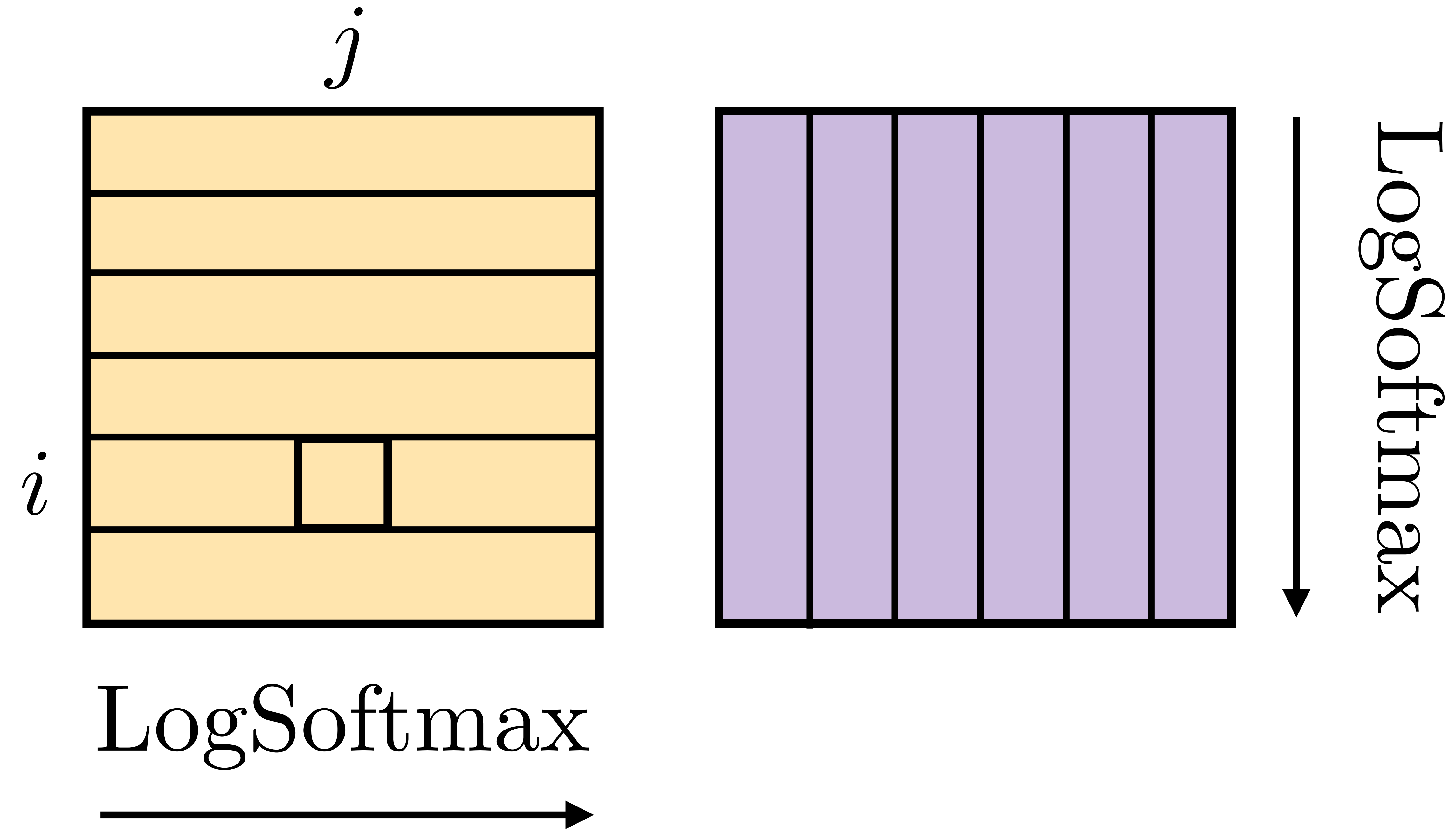
BERT Text Embeddings

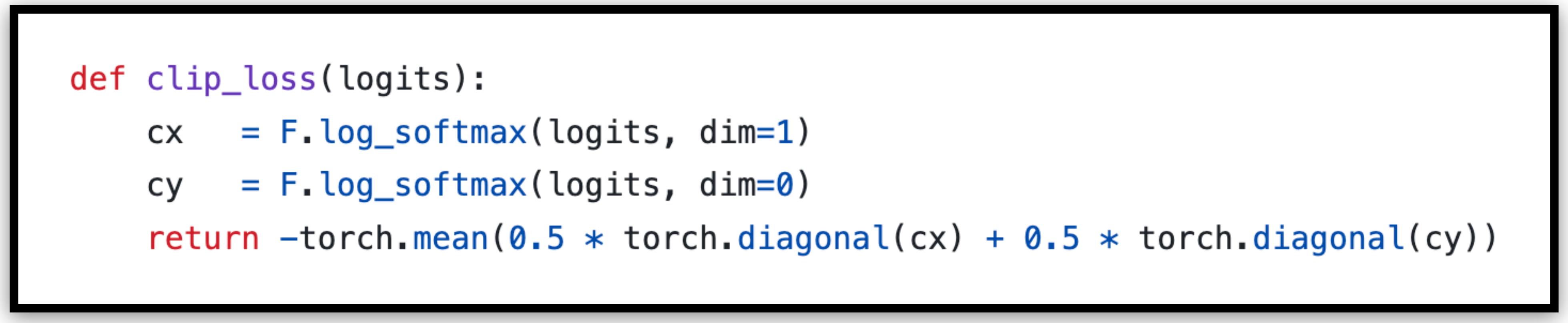
GPT-2 Text Embeddings

CIFAR-10

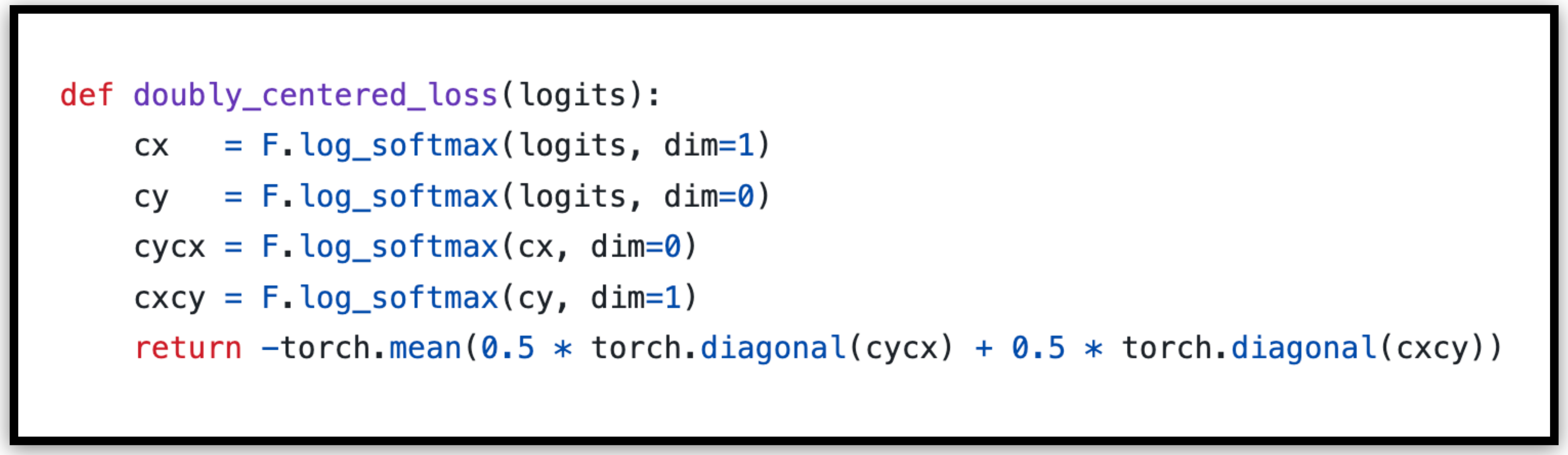
CIFAR-100

STL-10





```
def clip_loss(logits):
    cx   = F.log_softmax(logits, dim=1)
    cy   = F.log_softmax(logits, dim=0)
    return -torch.mean(0.5 * torch.diagonal(cx) + 0.5 * torch.diagonal(cy))
```

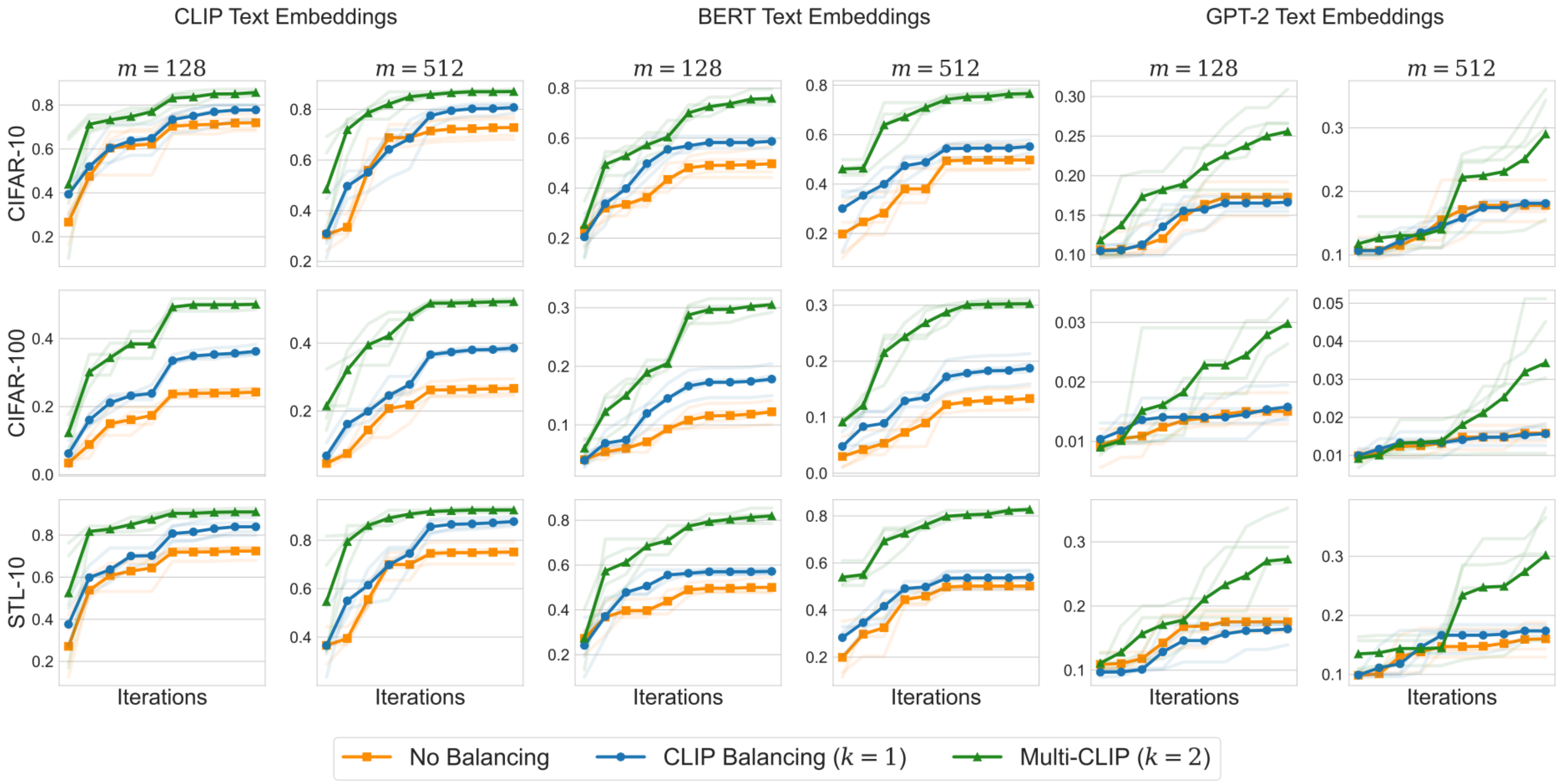


```
def doubly_centered_loss(logits):
    cx   = F.log_softmax(logits, dim=1)
    cy   = F.log_softmax(logits, dim=0)
    cycx = F.log_softmax(cx, dim=0)
    cxcy = F.log_softmax(cy, dim=1)
    return -torch.mean(0.5 * torch.diagonal(cycx) + 0.5 * torch.diagonal(cxcy))
```

No Balancing — CLIP Balancing ($k = 1$) — Multi-CLIP ($k = 2$)

Increasing the number of iterations results in zero-shot accuracy gains!

Increasing the number of iterations results in zero-shot accuracy gains!



Conclusion

Three Ingredients of Success

Pre-Training Data

Self-Supervised Learning Objective

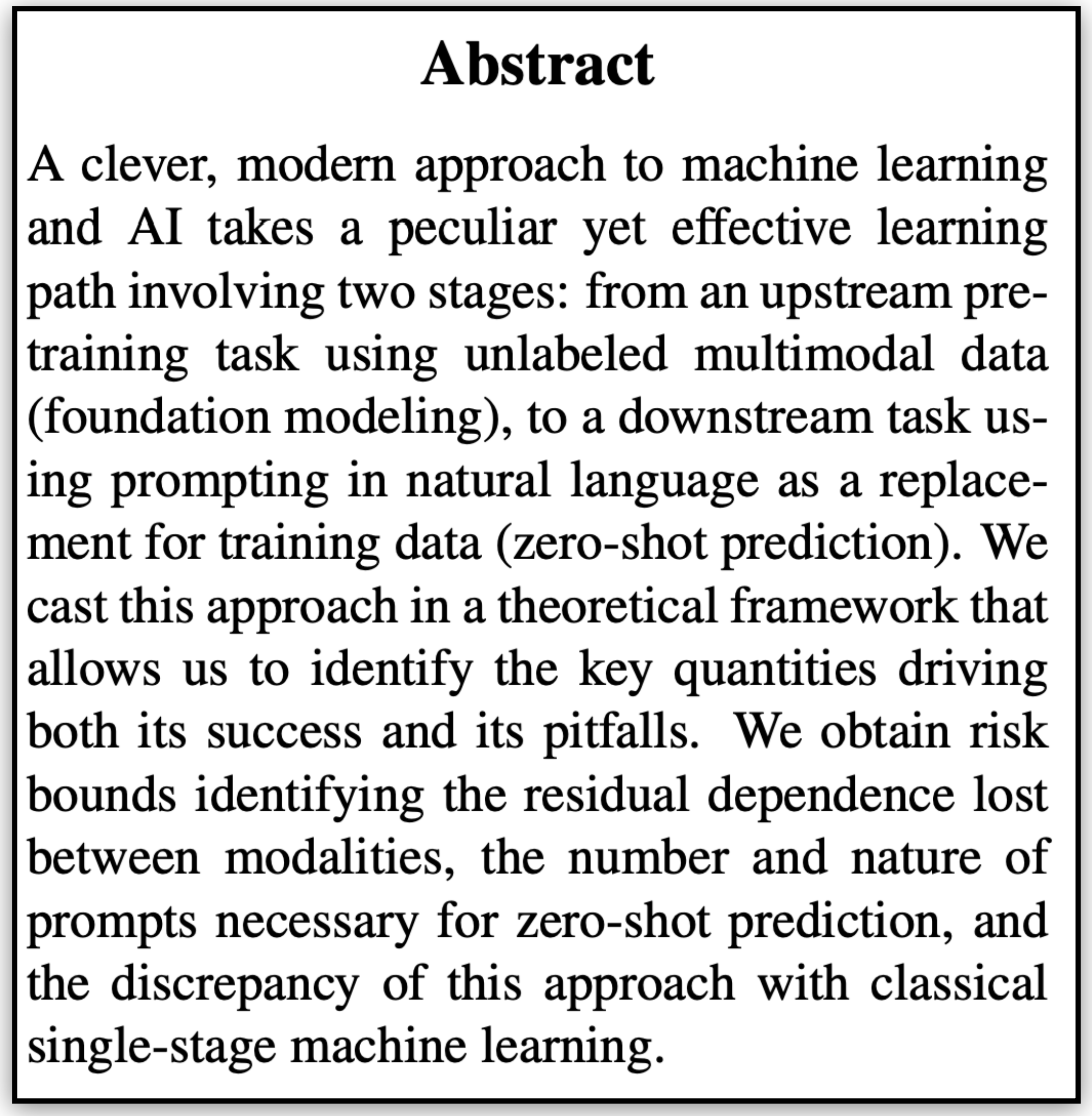
Prompting/ Pseudo-Captioning

What is the effect of common multimodal data curation methods on pre-training/downstream performance?

How do we interpret the CLIP objective (large batch limit, etc.) and improve it?

When can prompt-based zero-shot prediction match the performance of supervised learning?

From Pre-Training Foundation Models to Zero-Shot Prediction: Learning Paths, Prompt Complexity, and Residual Dependence



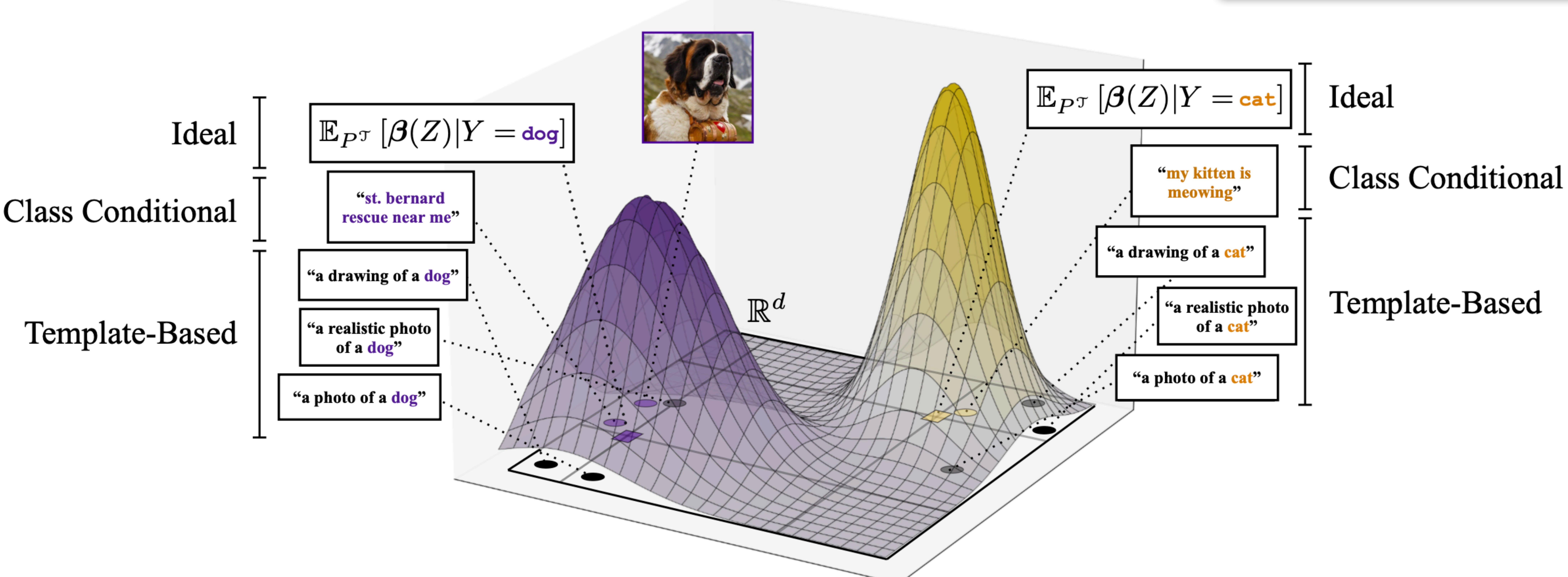
Abstract

A clever, modern approach to machine learning and AI takes a peculiar yet effective learning path involving two stages: from an upstream pre-training task using unlabeled multimodal data (foundation modeling), to a downstream task using prompting in natural language as a replacement for training data (zero-shot prediction). We cast this approach in a theoretical framework that allows us to identify the key quantities driving both its success and its pitfalls. We obtain risk bounds identifying the residual dependence lost between modalities, the number and nature of prompts necessary for zero-shot prediction, and the discrepancy of this approach with classical single-stage machine learning.

What is the entire pipeline estimating?

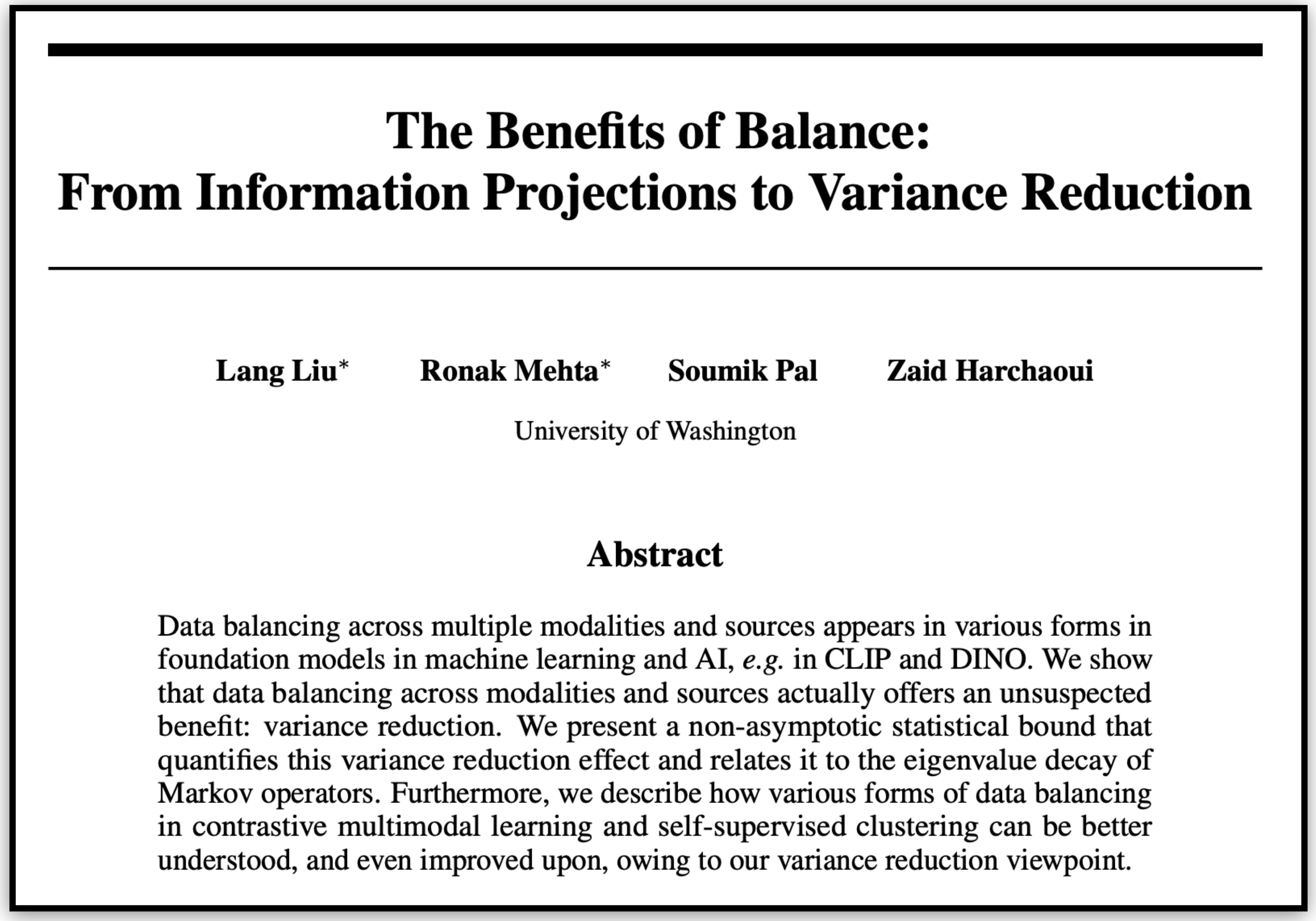
What is theoretically "ideal" prompting?

How close can this get to Bayes optimal performance?



Ideal
Class Conditional
Template-Based
$\mathbb{E}_{P^{\mathcal{T}}}[\boldsymbol{\beta}(Z)|Y = \texttt{dog}]$
"st. bernard rescue near me"
"a drawing of a dog"
"a realistic photo of a dog"
"a photo of a dog"
$\mathbb{R}^d$
$\mathbb{E}_{P^{\mathcal{T}}}[\boldsymbol{\beta}(Z)|Y = \texttt{cat}]$
"my kitten is meowing"
"a drawing of a cat"
"a realistic photo of a cat"
"a photo of a cat"
Ideal
Class Conditional
Template-Based

Reproducibility



The Benefits of Balance: From Information Projections to Variance Reduction

Lang Liu* Ronak Mehta* Soumik Pal Zaid Harchaoui

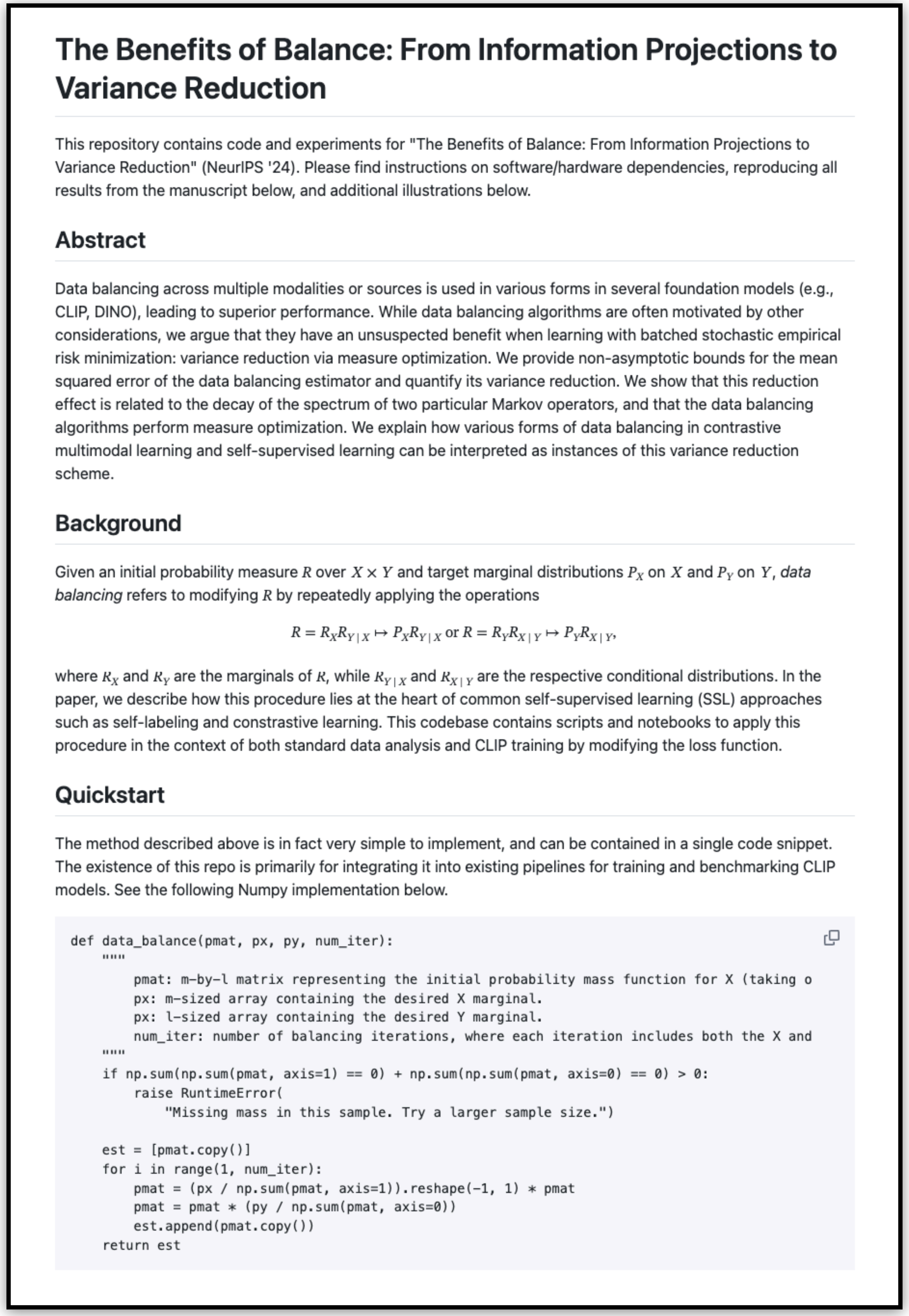
University of Washington

Abstract

Data balancing across multiple modalities and sources appears in various forms in foundation models in machine learning and AI, *e.g.* in CLIP and DINO. We show that data balancing across modalities and sources actually offers an unsuspected benefit: variance reduction. We present a non-asymptotic statistical bound that quantifies this variance reduction effect and relates it to the eigenvalue decay of Markov operators. Furthermore, we describe how various forms of data balancing in contrastive multimodal learning and self-supervised clustering can be better understood, and even improved upon, owing to our variance reduction viewpoint.

NeurIPS '24

SCAN ME



The Benefits of Balance: From Information Projections to Variance Reduction

This repository contains code and experiments for "The Benefits of Balance: From Information Projections to Variance Reduction" (NeurIPS '24). Please find instructions on software/hardware dependencies, reproducing all results from the manuscript below, and additional illustrations below.

Abstract

Data balancing across multiple modalities or sources is used in various forms in several foundation models (e.g., CLIP, DINO), leading to superior performance. While data balancing algorithms are often motivated by other considerations, we argue that they have an unsuspected benefit when learning with batched stochastic empirical risk minimization: variance reduction via measure optimization. We provide non-asymptotic bounds for the mean squared error of the data balancing estimator and quantify its variance reduction. We show that this reduction effect is related to the decay of the spectrum of two particular Markov operators, and that the data balancing algorithms perform measure optimization. We explain how various forms of data balancing in contrastive multimodal learning and self-supervised learning can be interpreted as instances of this variance reduction scheme.

Background

Given an initial probability measure R over $X \times Y$ and target marginal distributions P_X on X and P_Y on Y, *data balancing* refers to modifying R by repeatedly applying the operations

$$R = R_X R_{Y \mid X} \mapsto P_X R_{Y \mid X} \text{ or } R = R_Y R_{X \mid Y} \mapsto P_Y R_{X \mid Y},$$

where R_X and R_Y are the marginals of R, while $R_{Y \mid X}$ and $R_{X \mid Y}$ are the respective conditional distributions. In the paper, we describe how this procedure lies at the heart of common self-supervised learning (SSL) approaches such as self-labeling and constrastive learning. This codebase contains scripts and notebooks to apply this procedure in the context of both standard data analysis and CLIP training by modifying the loss function.

Quickstart

The method described above is in fact very simple to implement, and can be contained in a single code snippet. The existence of this repo is primarily for integrating it into existing pipelines for training and benchmarking CLIP models. See the following Numpy implementation below.

```
def data_balance(pmat, px, py, num_iter):
    """
        pmat: m-by-l matrix representing the initial probability mass function for X (taking o
        px: m-sized array containing the desired X marginal.
        px: l-sized array containing the desired Y marginal.
        num_iter: number of balancing iterations, where each iteration includes both the X and
    """
    if np.sum(np.sum(pmat, axis=1) == 0) + np.sum(np.sum(pmat, axis=0) == 0) > 0:
        raise RuntimeError(
            "Missing mass in this sample. Try a larger sample size.")

    est = [pmat.copy()]
    for i in range(1, num_iter):
        pmat = (px / np.sum(pmat, axis=1)).reshape(-1, 1) * pmat
        pmat = pmat * (py / np.sum(pmat, axis=0))
        est.append(pmat.copy())
    return est
```

Thank you!

Appendix

Theorem (Liu, M., Pal, Harchaoui)

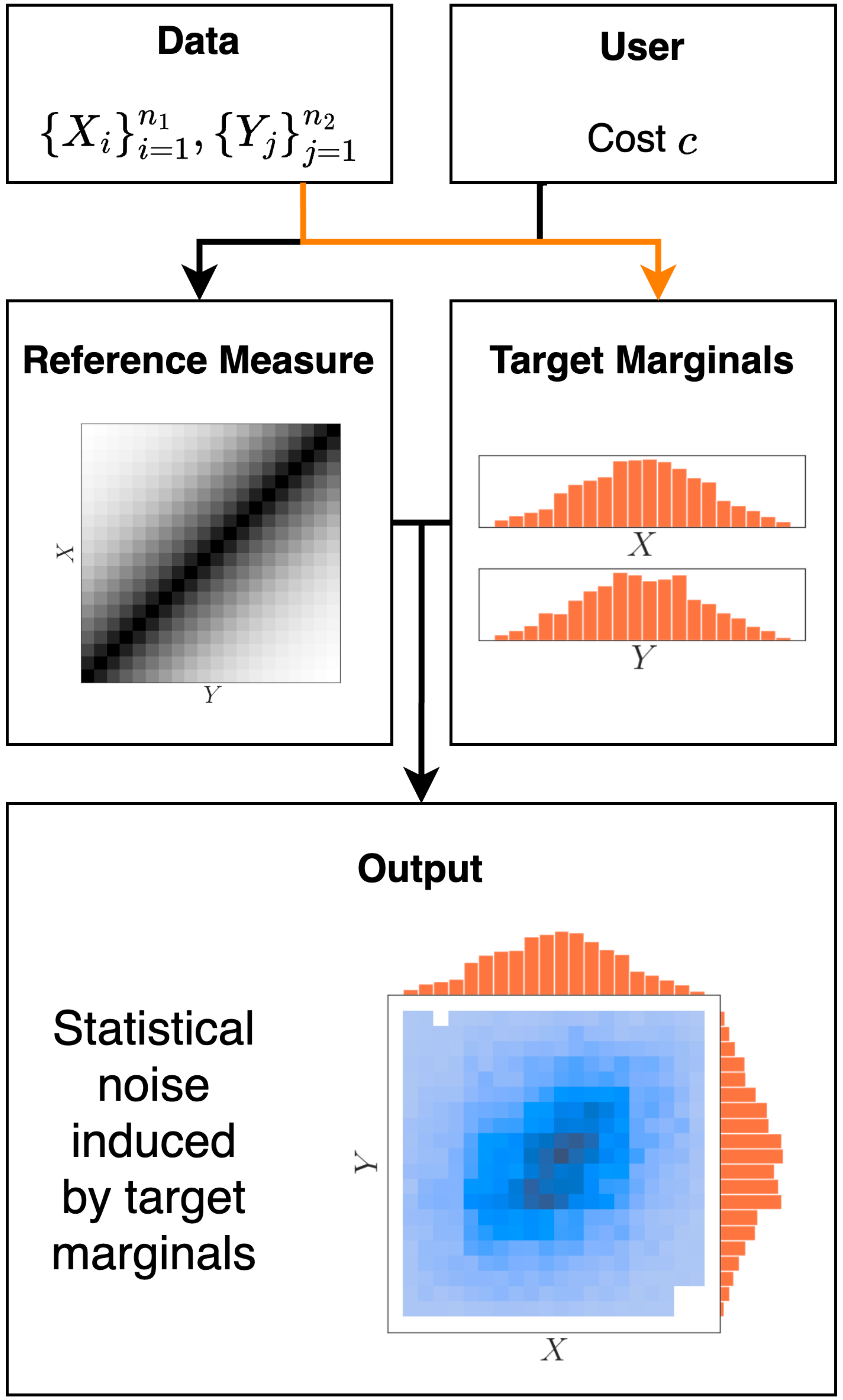
$$\mathbb{E}_{P^n}\left[(P_n^{(k)}(h) - P(h))^2\right] = \frac{\mathbb{V}\mathrm{ar}(\mathcal{C}_Z\mathcal{C}_X \ldots \mathcal{C}_Z\mathcal{C}_X h)}{n} + \boxed{\tilde{O}\left(\frac{k^6}{n^{3/2}}\right)}$$

$$[P_n^{(\ell)} - P_n^{(\ell-1)}](\mathcal{C}_\ell \ldots \mathcal{C}_k h) = \sum_{\boldsymbol{x},\boldsymbol{z}} \left[\frac{P_X(\boldsymbol{x})}{P_{n,X}^{(\ell-1)}(\boldsymbol{x})} - 1\right] \cdot [\mathcal{C}_\ell \ldots \mathcal{C}_k h](\boldsymbol{x},\boldsymbol{z}) P_n^{(\ell-1)}(\boldsymbol{x},\boldsymbol{z}).$$

$$\underbrace{\sum_{\ell=1}^k [P_n^{(\ell)} - P_n^{(\ell-1)}](\mathcal{C}_\ell \ldots \mathcal{C}_k h)}.$$

Higher-Order Term

Entropy-Regularized Optimal Transport



Data

$\{X_i\}_{i=1}^{n_1}, \{Y_j\}_{j=1}^{n_2}$

User

Cost c

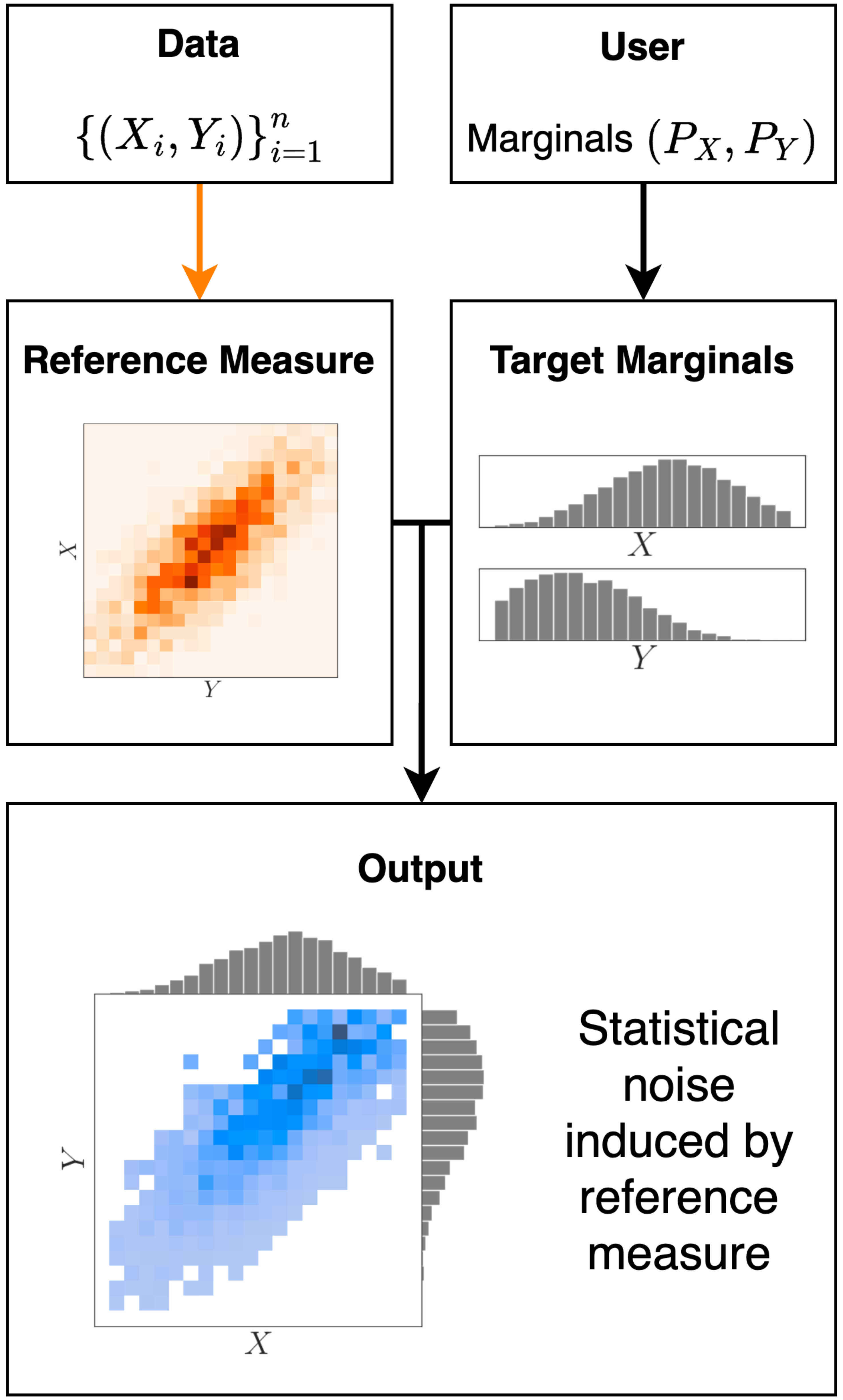
Reference Measure

Target Marginals

Output

Statistical noise induced by target marginals

Marginal Rebalanced Estimation



Data

$\{(X_i, Y_i)\}_{i=1}^{n}$

User

Marginals (P_X, P_Y)

Reference Measure

Target Marginals

Output

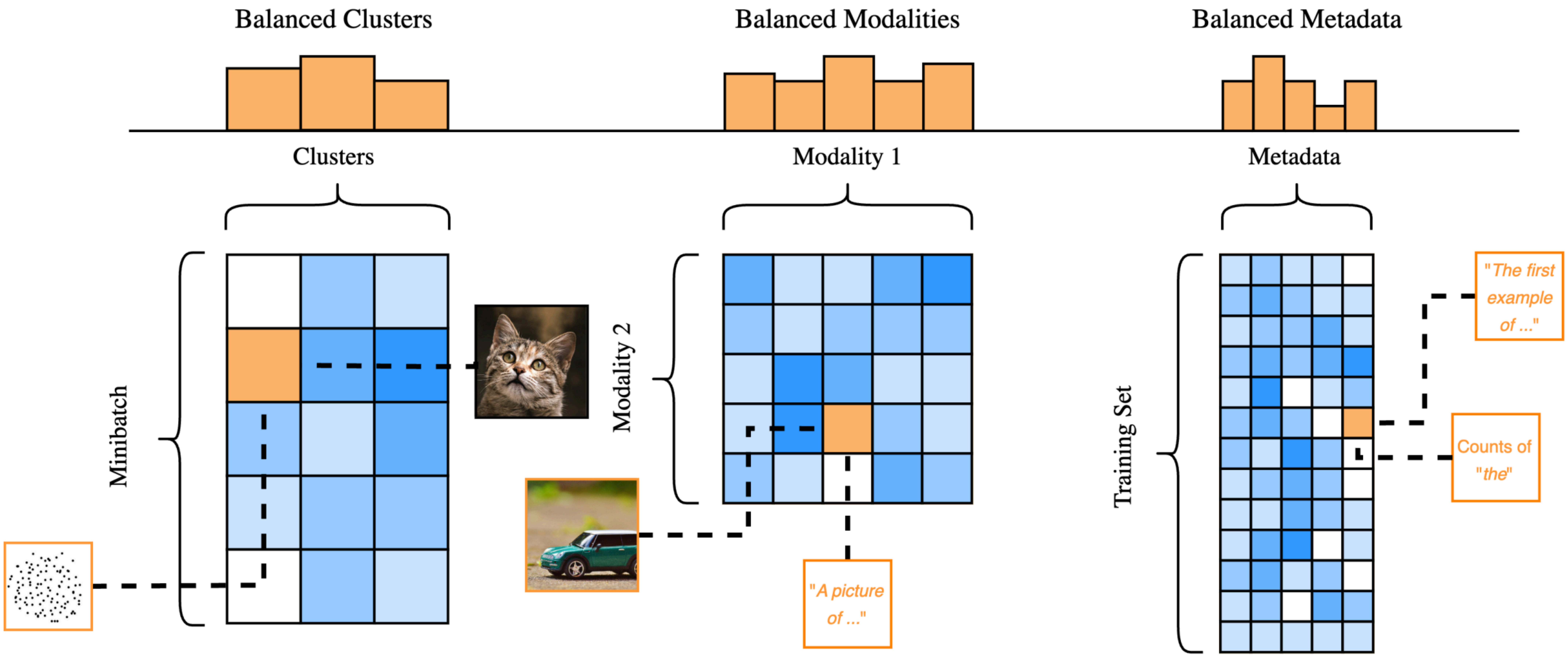
Statistical noise induced by reference measure

Assumption 4.6.1. There exist fixed probability mass functions $\hat{P}_X$ and $\hat{P}_Z$ for some $\varepsilon \in [0, 1)$,

$$\hat{P}_{X,\varepsilon} = (1-\varepsilon)P_X + \varepsilon \hat{P}_X \text{ and } \hat{P}_{Z,\varepsilon} = (1-\varepsilon)P_Z + \varepsilon \hat{P}_Z.$$

Theorem 4.6.1. *Let Asm. 4.6.1 be true with error $\varepsilon \in [0, 1)$. For a sequence of rebalanced distributions $(\hat{P}_n^{(k)})_{k \geq 1}$, there exists an absolute constant $C > 0$ such that when $n \geq C[\log_2(2n/\hat{p}_{\star,\varepsilon}) + m \log(n+1)]/\min\{p_\star, \hat{p}_{\star,\varepsilon}\}^2$, we have that*

$$\begin{aligned}
&\mathbb{E}_P\left[\left(\hat{P}_n^{(k)}(h) - P(h)\right)^2 \mathbb{1}_{\mathcal{S}}\right] + \mathbb{E}_P\left[(P_n(h) - P(h))^2 \mathbb{1}_{\mathcal{S}^c}\right] \leq \frac{\sigma_k^2}{n} + \tilde{O}\left(\frac{k^6}{n^{3/2}}\right) \\
&+ \tilde{O}\left(\frac{k^4}{\hat{p}_{\star,\varepsilon}^2}\left(\sqrt{\frac{1}{n}\log\frac{1}{1-\varepsilon}} + \log\frac{1}{1-\varepsilon}\right)\left[\frac{k^2}{\hat{p}_{\star,\varepsilon}^2}\left(\sqrt{\frac{1}{n}\log\frac{1}{1-\varepsilon}} + \log\frac{1}{1-\varepsilon} + \frac{1}{n}\right) + \frac{1}{\sqrt{n}}\right]\right) \\
&+ \tilde{O}\left(k^2\left[\sqrt{\varepsilon}\left(\frac{\hat{p}_{\star,\varepsilon}^4}{n^4} + \frac{1}{\sqrt{n}} + \frac{\hat{p}_{\star,\varepsilon}^2 k}{n^4}\left(n + \frac{k^2}{\hat{p}_{\star,\varepsilon}^2}\right) + \frac{k^2}{\hat{p}_{\star,\varepsilon}^2}\left[\frac{1}{n} + \sqrt{\frac{1}{n}\log\frac{1}{1-\varepsilon}} + \log\frac{1}{1-\varepsilon}\right]\right) + \varepsilon\right]\right).
\end{aligned}$$

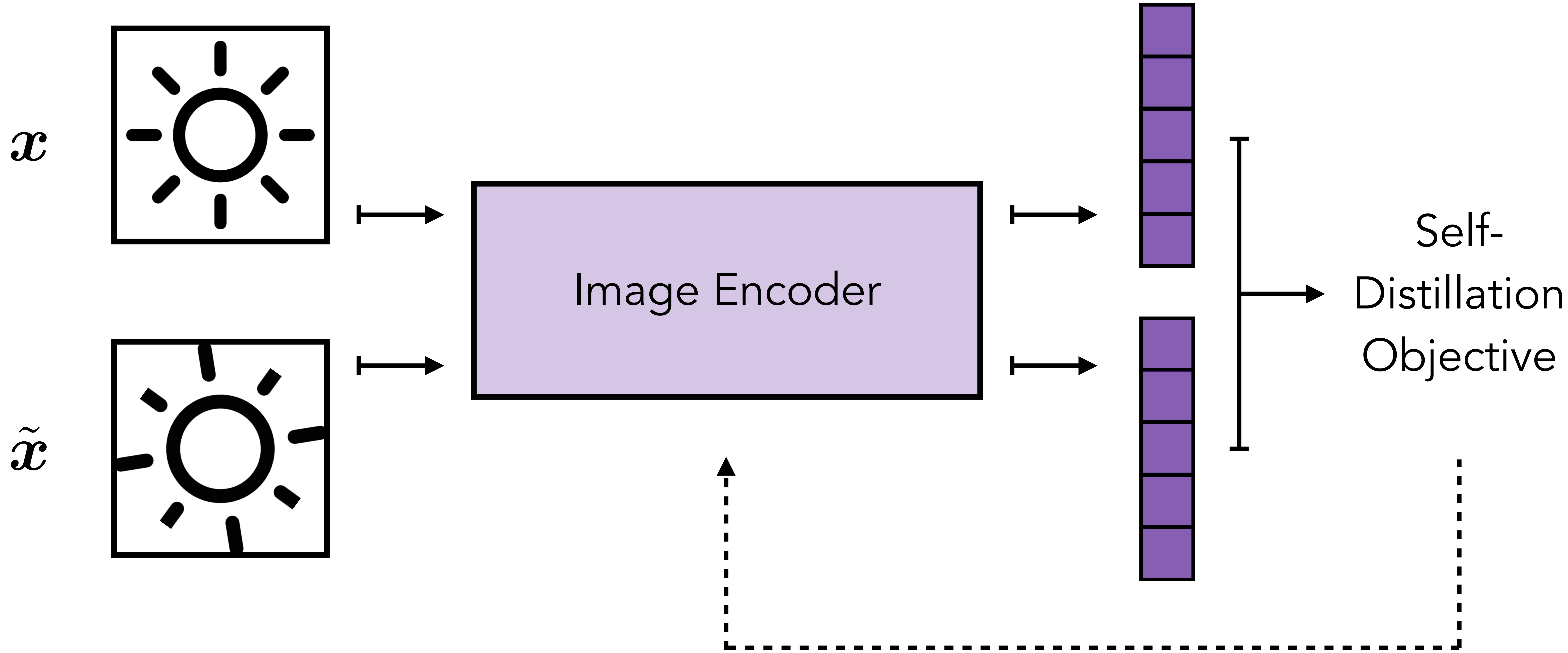


Self-Supervised Clustering

Contrastive Learning

Metadata Curation

Pre-Training: Self-Supervised Learning


x
$\tilde{x}$
Image Encoder
Self-Distillation Objective