



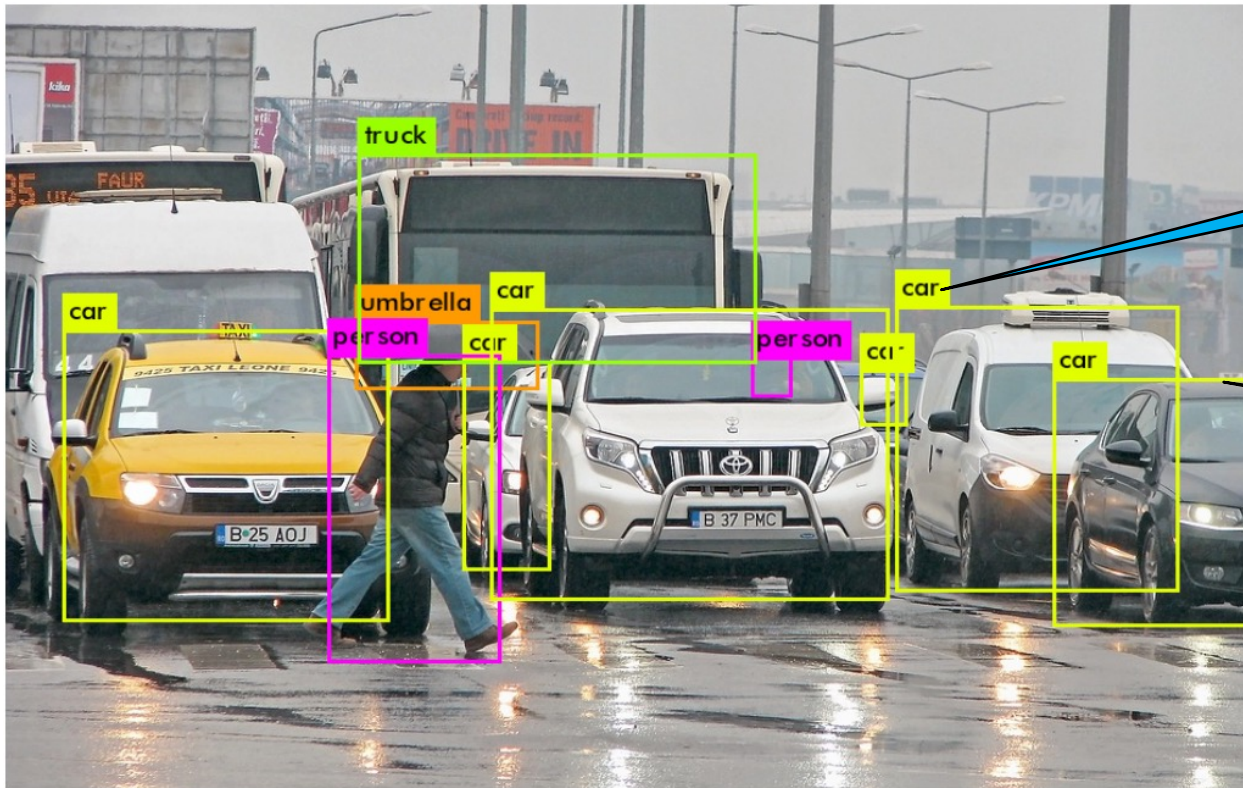
Ailio

Non-Contrastive Self-Supervised Learning

29.11.2022

Julien Marteen Akay | jakay@ailio.de

Deep Learning - Beispiel



Klassifikation

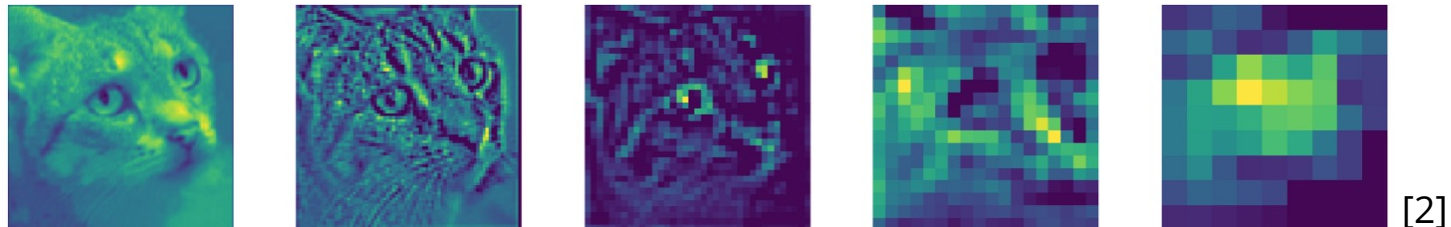
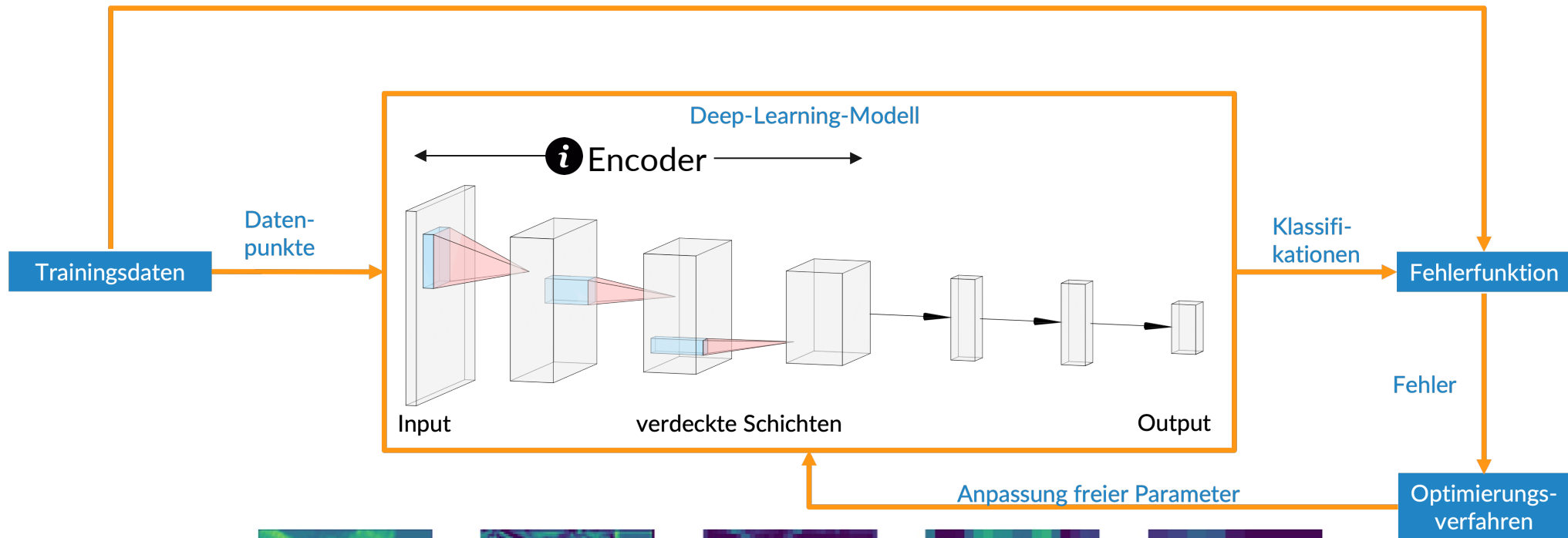
Regression

[1]

Deep Learning

Supervised Learning: „Lernen mit Labels“

Labels



zurück zur Realität

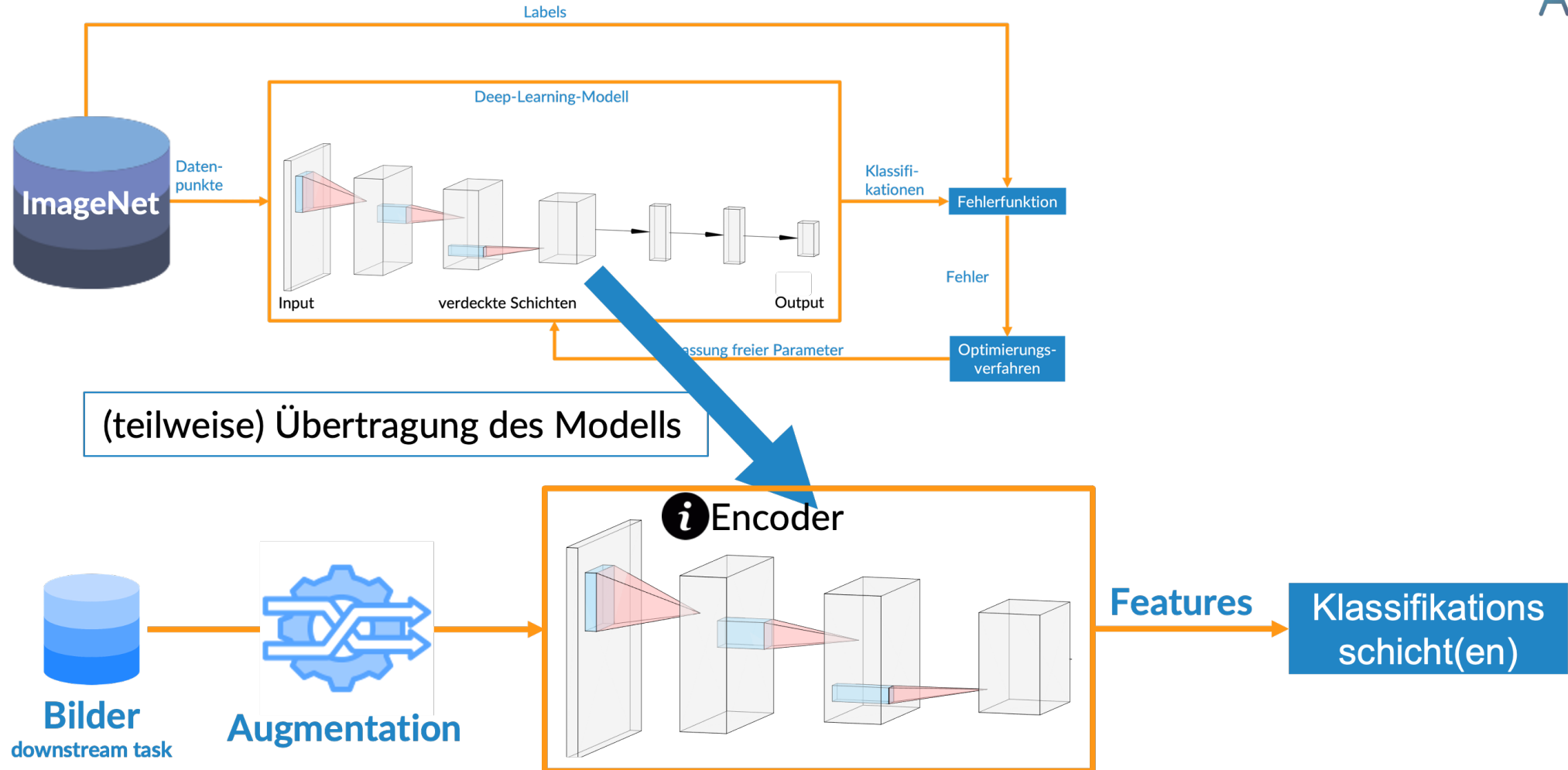
Anforderungen

- ❖ viele Daten
- ❖ Labels
- ❖ Hardware
- ❖ Zeit

Gegebenheiten

- ❖ wenige Daten
- ❖ entsprechend wenige Labels

Transfer Learning



Transfer Learning



Encoder lernen

übertragen

Aufgabe lösen

- ❖ Supervised Learning
- ❖ aufgabenspezifisch
- ❖ Domäne A
- ❖ **i** Feature Learning

- ❖ aufgabenspezifischer Encoder
- ❖ Features/Wissen aus Domäne A

- ❖ Supervised Learning
- ❖ aufgabenspezifisch
- ❖ Daten aus Domäne B
- ❖ Features aus Domäne A
- ❖ Lösung der Aufgabe

Transfer Learning

Encoder lernen

- ❖ Supervised Learning
- ❖ aufgabenspezifisch
- ❖ Domäne A
- ❖ **i** Feature Learning

übertragen

- ❖ aufgabenspezifischer Encoder
- ❖ Features/Wissen aus Domäne A

Aufgabe lösen

- ❖ Supervised Learning
- ❖ aufgabenspezifisch
- ❖ Daten aus Domäne B
- ❖ Features aus Domäne A
- ❖ Lösung der Aufgabe

Geht das besser?



Ailio

Self-Supervised Learning

„Füllen von Lücken“
Lernen ohne Labels

Contrastive SSL Computer Vision

Allgemein

- ❖ aufgabenagnostisch
→ keine Labels
- ❖ „Lücken füllen“

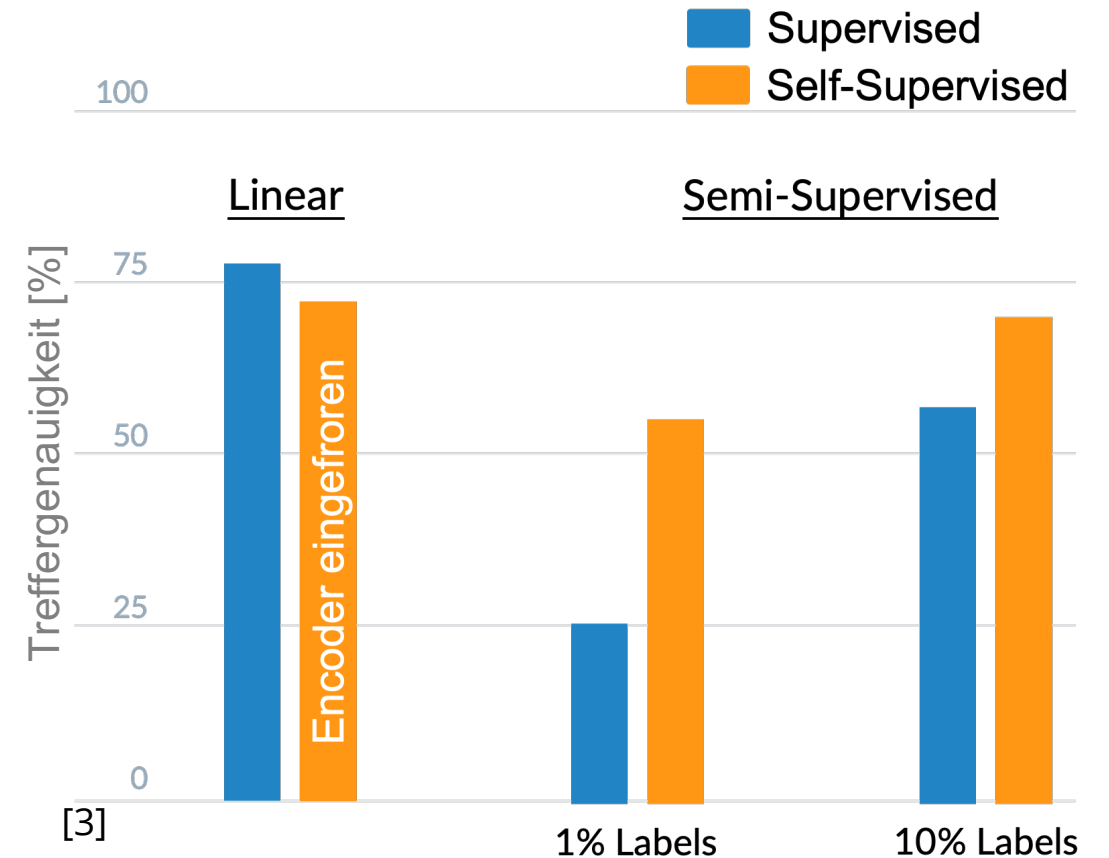
randomly
masked

A quick [MASK] fox jumps over the [MASK] dog

↓

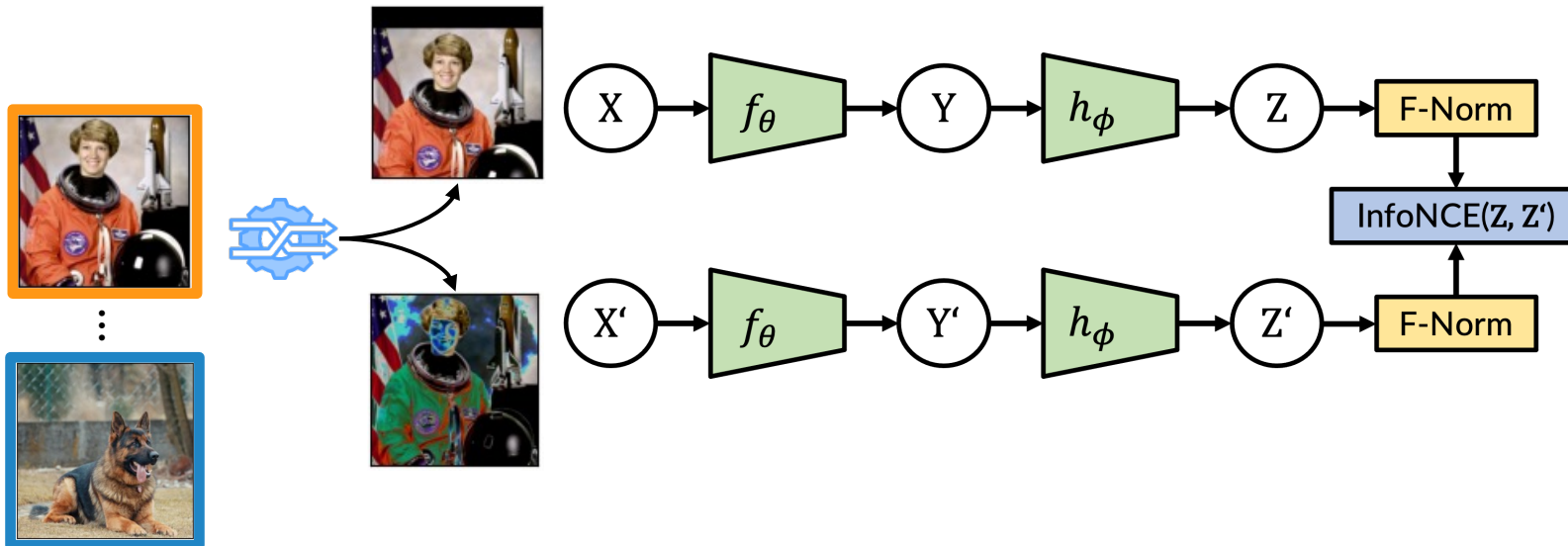
predict

A quick brown fox jumps over the lazy dog

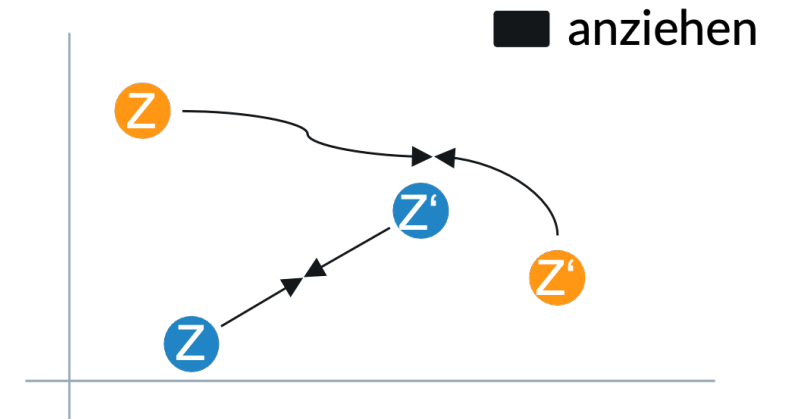


Contrastive SSL Computer Vision

Joint Embedding Architecture



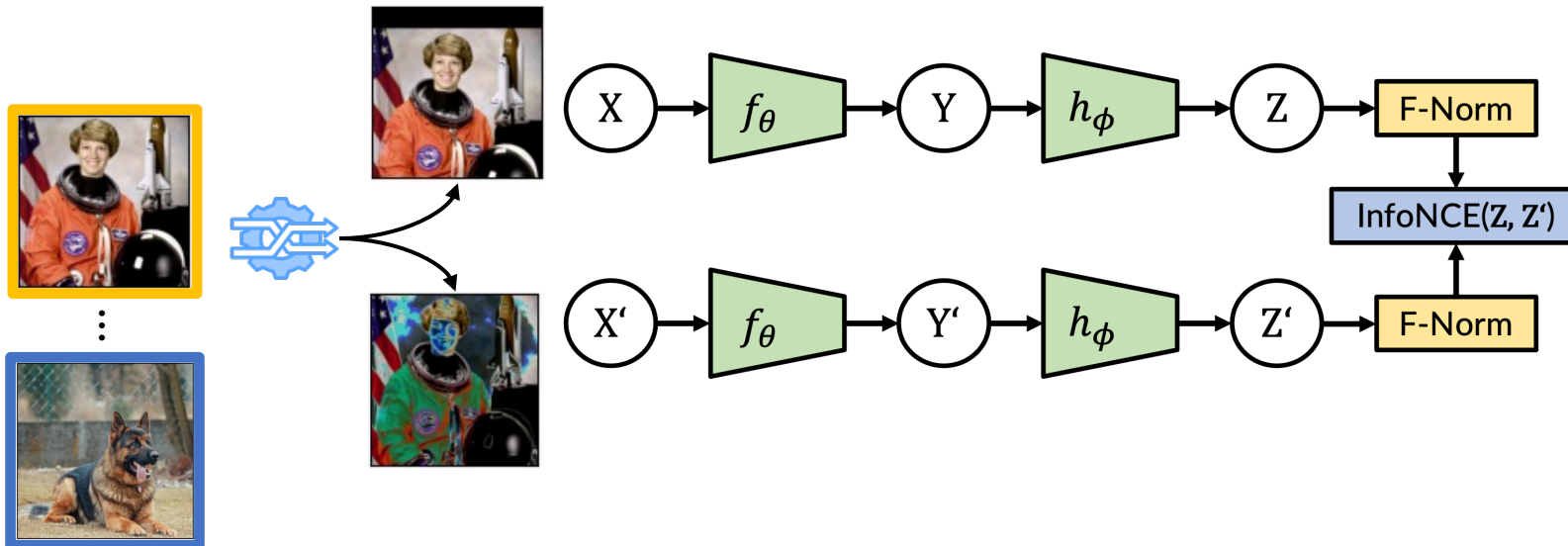
Contrast



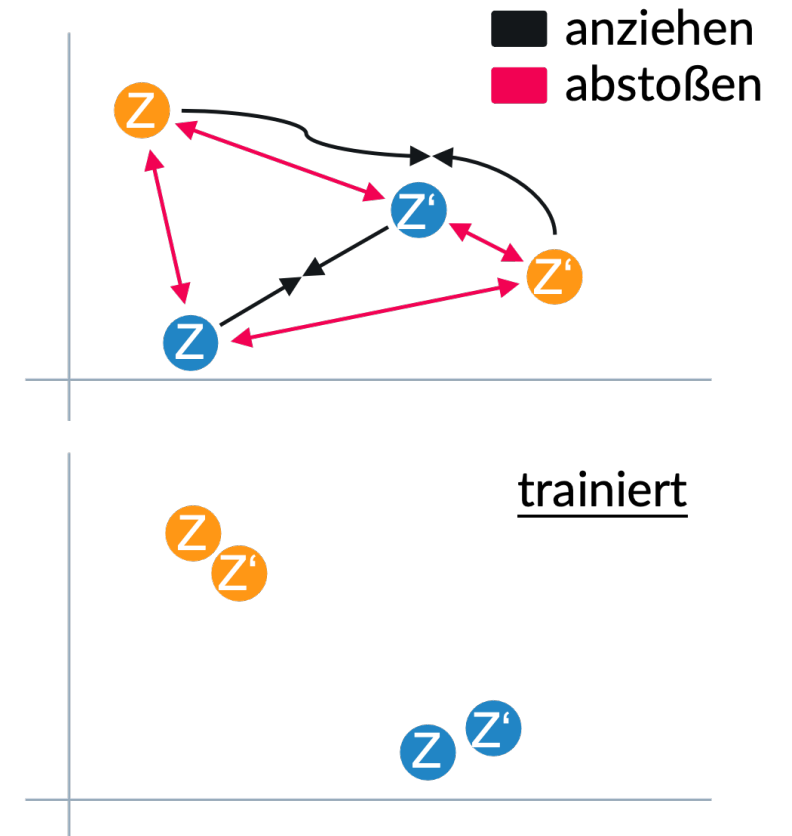
[3, 4]

Contrastive SSL Computer Vision

Joint Embedding Architecture

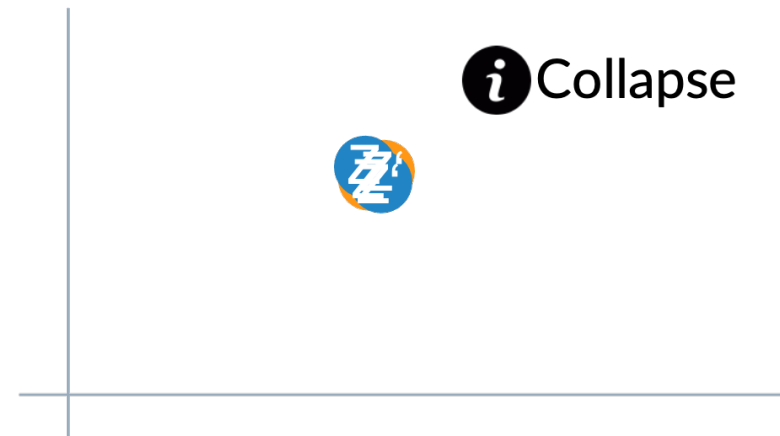
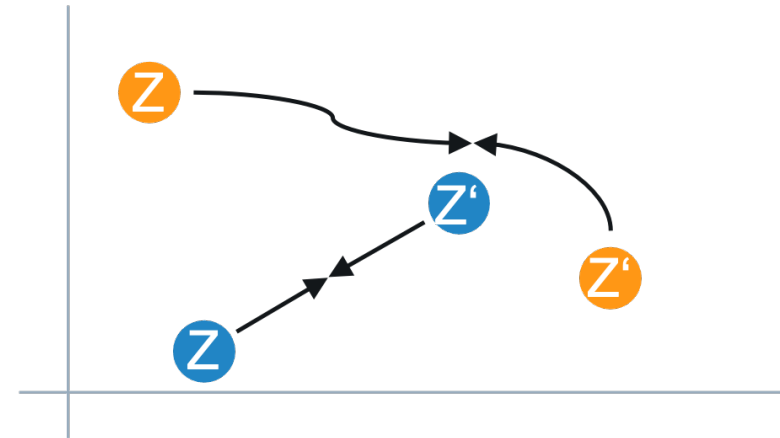


Contrast



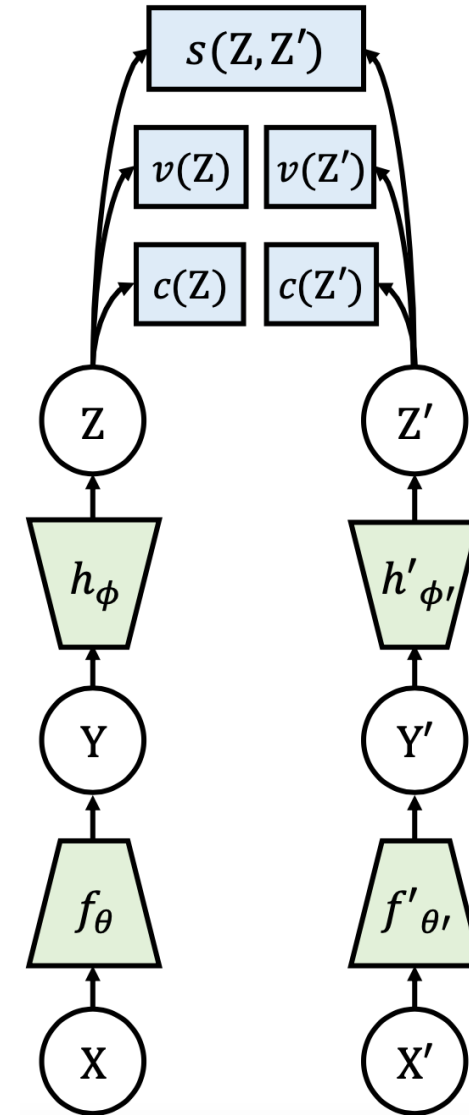
Non-Contrastive SSL - Collapse

- ❖ kein Kontrast
- ⚠ triviale Lösung möglich
- ⚠ Features sind nutzlos
- ❖ Inputs werden „ignoriert“
- ❖ konstante Embeddings



Variance-Invariance-Covariance-Regularization

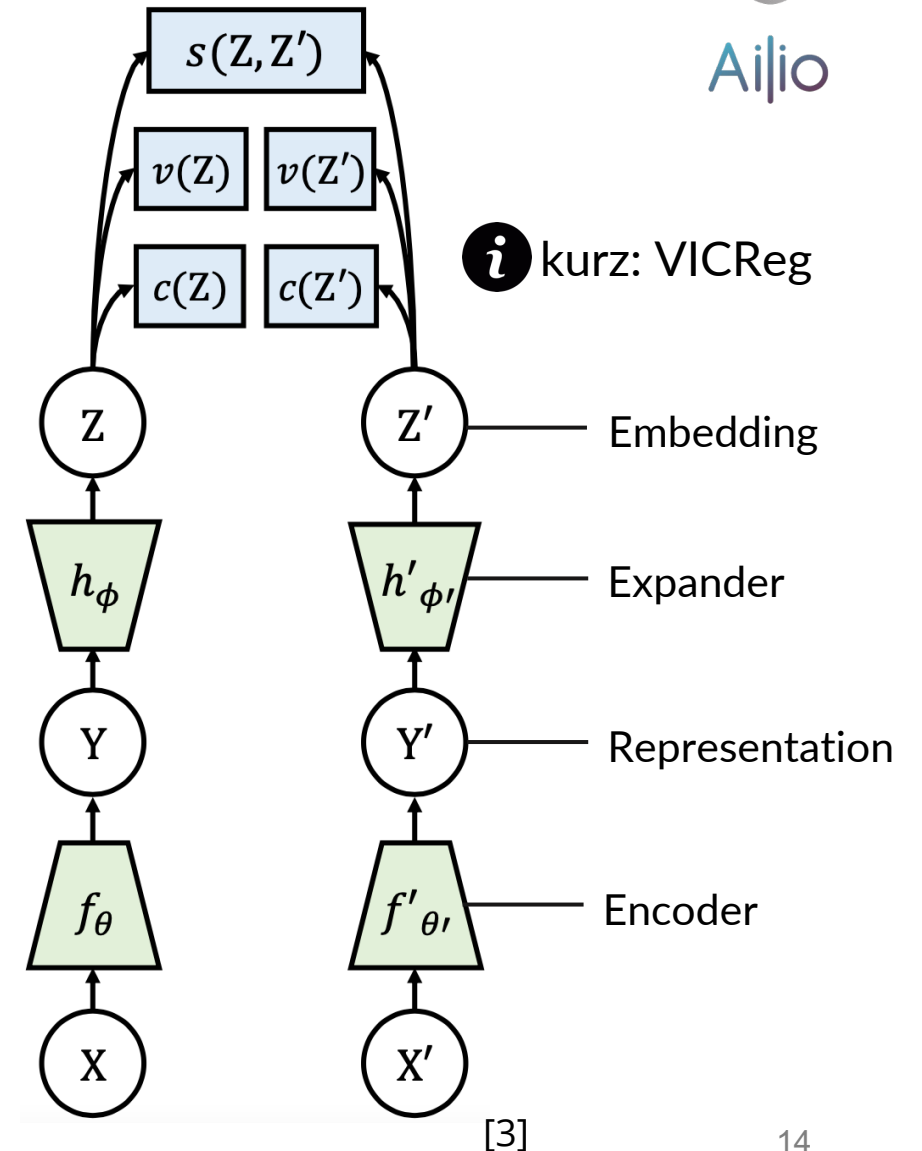
- ❖ Non-Contrastive SSL
- ❖ wichtigste Elemente
 - Fehlerfunktion
 - Augmentierung
- ❖ keine speziellen „Tricks“
 - Momentum Encoder
 - Stop Gradient
 - Clustering
 - ...



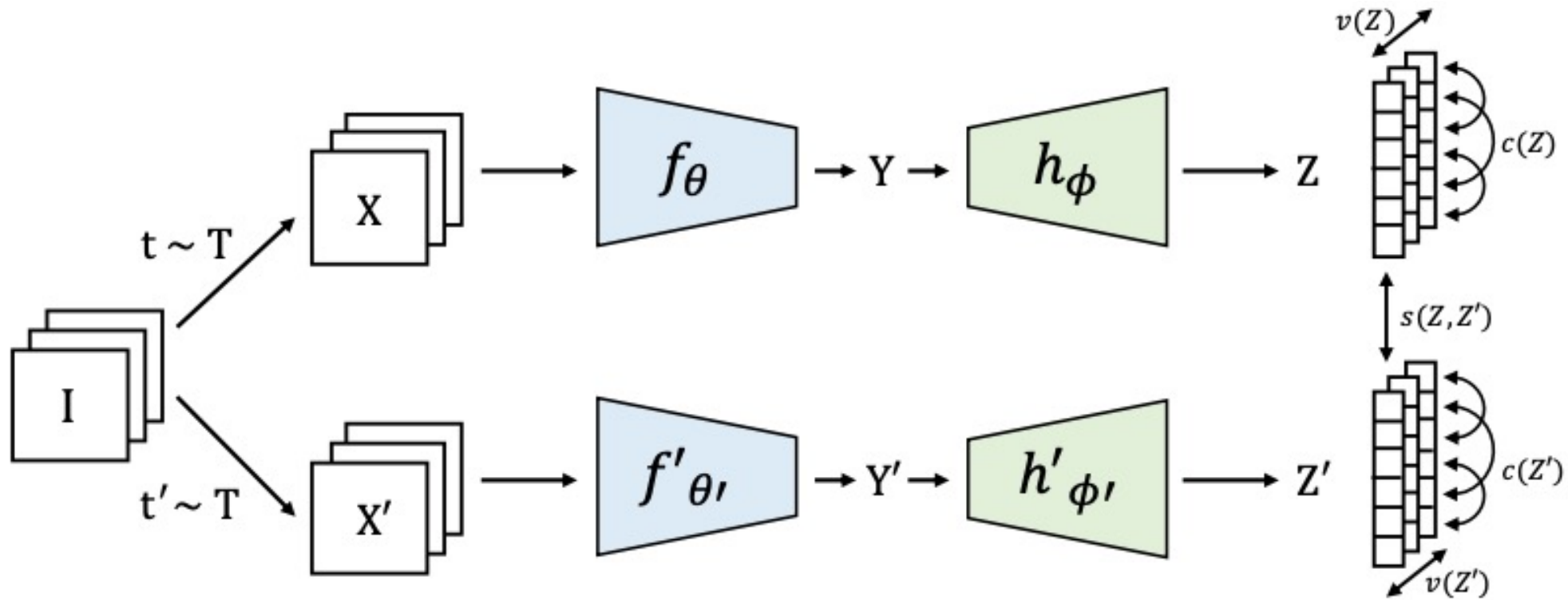
[3]

Variance-Invariance-Covariance-Regularization

- ❖ Non-Contrastive SSL
- ❖ wichtigste Elemente
 - Fehlerfunktion
 - Augmentierung
- ❖ keine speziellen „Tricks“
 - Momentum Encoder
 - Stop Gradient
 - Clustering
 - ...



VICReg



Fehlerfunktion - Übersicht

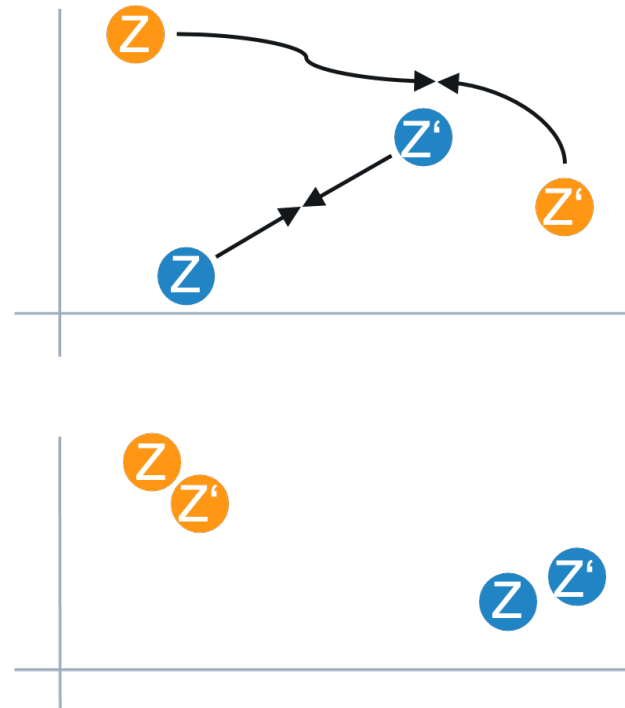
Variance

- ❖ hält Varianz aufrecht
- ❖ Ohne Variance-Regularization
→ konstante Repräsentationen

i VI-Reg funktioniert auch

Invariance

- ❖ minimiert Abstand
- ❖ Mean-Squared-Error



Covariance

- ❖ Dekorrelation
- ❖ Embedding-Variablen mit möglichst wenig redundanten Infos
- ❖ verhindert informativen Collapse

Fehlerfunktion - Variance

Variance

- ❖ hält Varianz aufrecht
- ❖ Ohne Variance-Regularization
→ konstante Repräsentationen

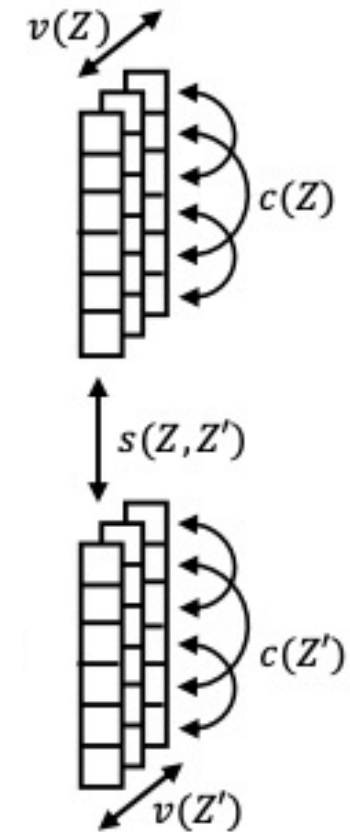
i VI-Reg funktioniert auch

$$v(Z) = \frac{1}{d} \sum_{j=1}^d \max(0, \gamma - S(z^j, \varepsilon))$$

$$S(x, \varepsilon) = \sqrt{\text{Var}(x) + \varepsilon}$$

$$\gamma = 1$$

i Hyperparameter



$$Z = \begin{pmatrix} z_1^1 & \dots & z_n^1 \\ \vdots & \ddots & \vdots \\ z_1^d & \dots & z_n^d \end{pmatrix}$$

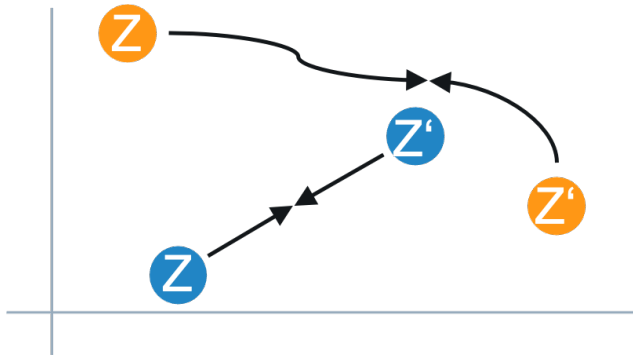
[3] analog für Z' **i**



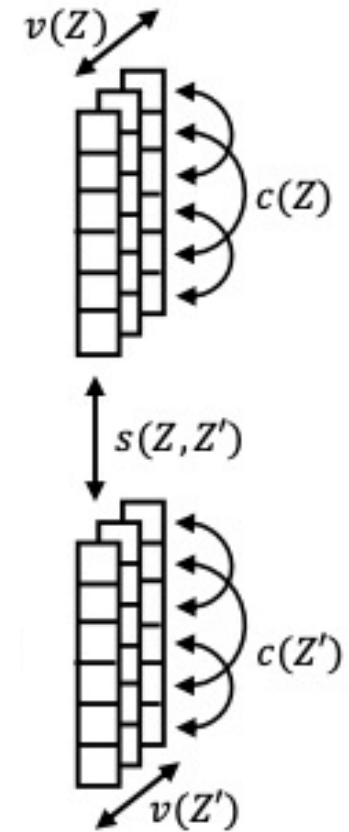
Fehlerfunktion - Invariance

Invariance

❖ minimiert Abstand



$$s(Z, Z') = \frac{1}{n} \sum_{i=1}^n \|z_i - z'_i\|_2^2$$



$$Z = \begin{pmatrix} z_1^1 & \dots & z_n^1 \\ \vdots & \ddots & \vdots \\ z_1^d & \dots & z_n^d \end{pmatrix}$$

[3] analog für Z' 



Fehlerfunktion - Covariance

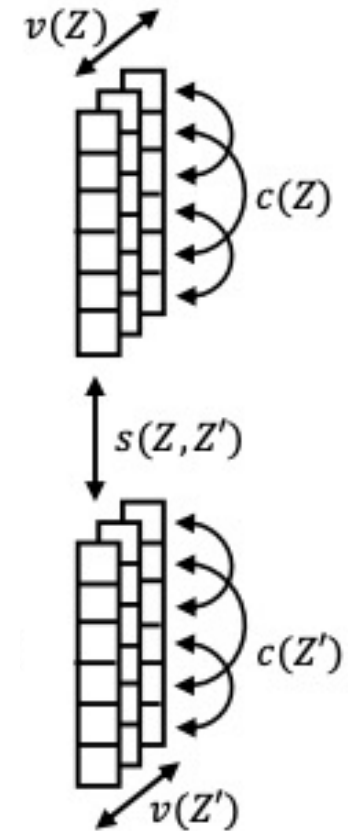
Covariance

- ❖ Dekorrelation
- ❖ Embedding-Variablen mit möglichst wenig redundanten Infos
- ❖ informativen Collapse verhindern

$$c(Z) = \frac{1}{d} \sum_{i \neq j} [C(Z)]_{i,j}^2$$

$$C(Z) = \frac{1}{n-1} \sum_{i=1}^n (z_i - \bar{z})(z_i - \bar{z})^T$$

$$\bar{z} = \frac{1}{n} \sum_{i=1}^n z_i$$



$$Z = \begin{pmatrix} z_1^1 & \dots & z_n^1 \\ \vdots & \ddots & \vdots \\ z_1^d & \dots & z_n^d \end{pmatrix}$$

[3] analog für Z' **i**

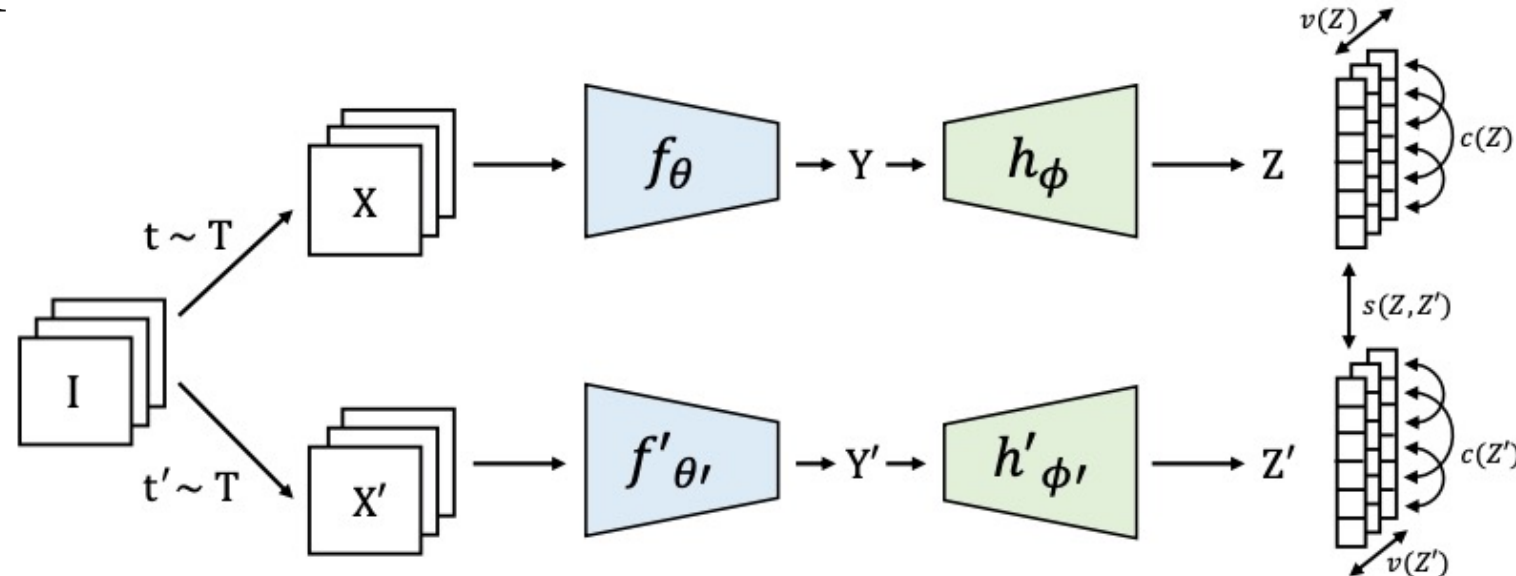
Fehlerfunktion - Gesamt

$$l(Z, Z') = \lambda s(Z, Z') + \mu[v(Z) + v(Z')] + \nu[c(Z) + c(Z')]$$

❖ Suche von Gewichten z.B. via Grid-Search

❖ $\lambda = \mu > \nu$

❖ $\nu = 1$





Ailio

Danke! Fragen?

Mehr zu Self-Supervised Learning, VICReg und World Models für autonome KI, gerne im nächsten Vortrag.

Quellen



- [1] K. Potdar und C. Pai, „A Convolutional Neural Network based Live Object Recognition System as Blind Aid“, *arXiv*, November 2018. Zugriff am: 29. November 2022. [Online]. Verfügbar unter: <https://arxiv.org/abs/1811.10399>
- [2] A. Dertat. „Applied Deep Learning - Part 4: Convolutional Neural Networks“. Medium. <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2#9722> (Zugriff am 28. November 2022).
- [3] A. Bardes, J. Ponce und Y. LeCun, „VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning“, in International Conference on Learning Representations, Virtual, 25. April 2022. Zugriff am: 29. November 2022. [Online]. Verfügbar unter: <https://arxiv.org/abs/2105.04906>
- [4] „Illustration of transforms — Torchvision main documentation“. PyTorch. https://pytorch.org/vision/main/auto_examples/plot_transforms.html - sphx-glr-auto-examples-plot-transforms-py (Zugriff am 28. November 2022).