Special Topics

Types of machine learning

	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Goal	**Predict** …from examples	**Describe** ...structure in data	**Strategize** learn by trial and error
Data	(x, y)	x	delayed feedback
Types	• Classification • Regression	• Density estimation • Clustering • Dimensionality reduction • Anomaly detection	• Model-free learning • Model-based learning

Special Topics

Semi-supervised learning

Self-supervised learning

Recommender systems

Other Practical Considerations

Special Topics

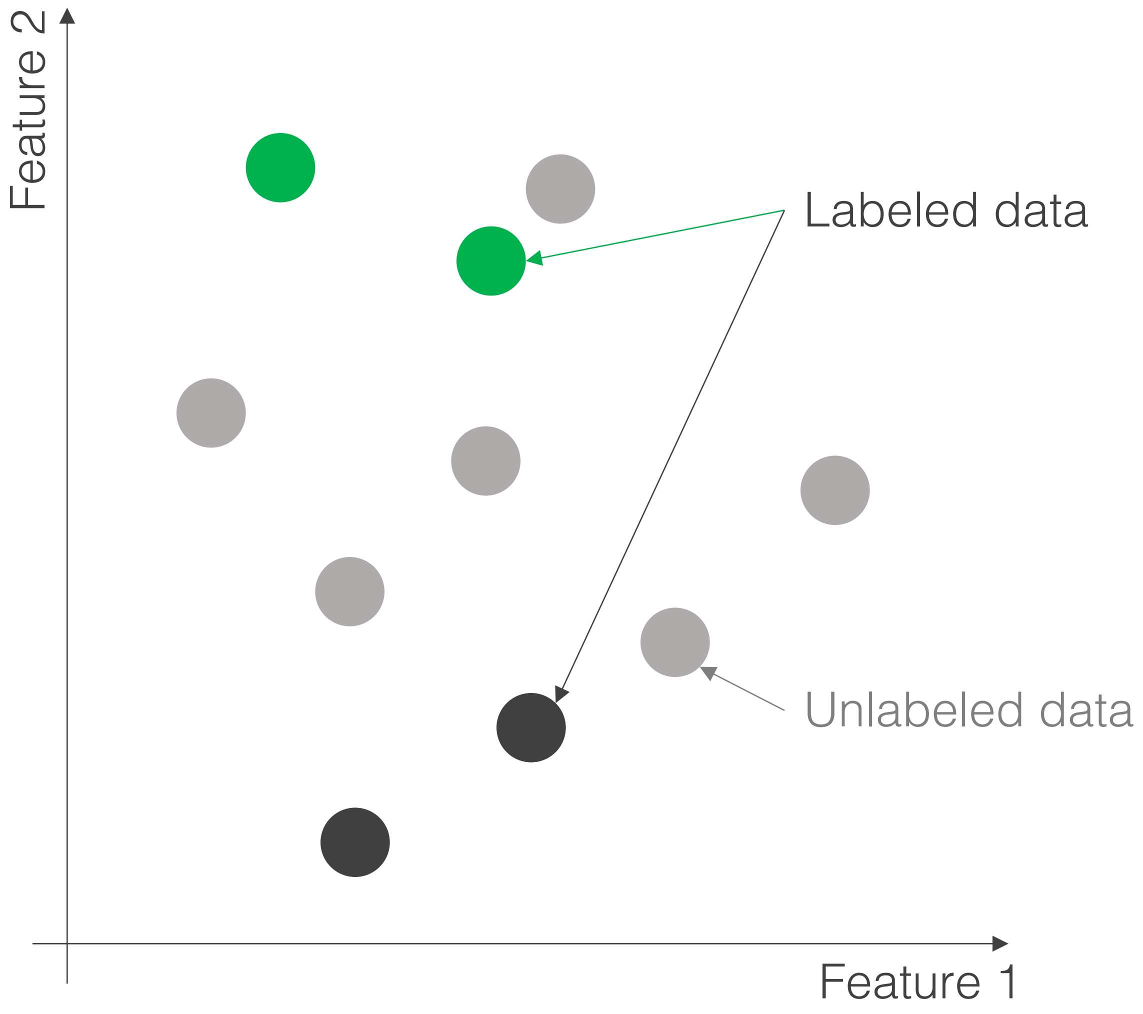
Semi-supervised learning

Self-supervised learning

Recommender systems

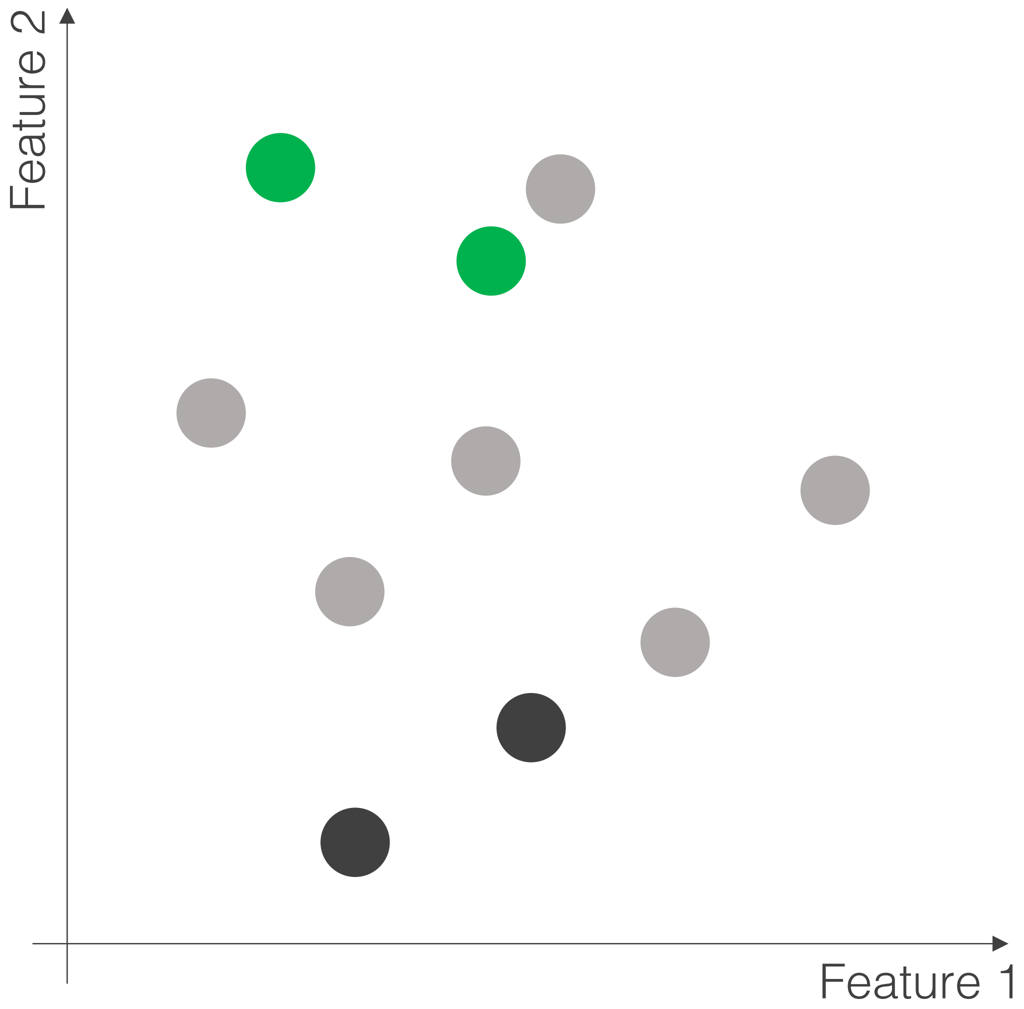
Other Practical Considerations

Semi-supervised learning

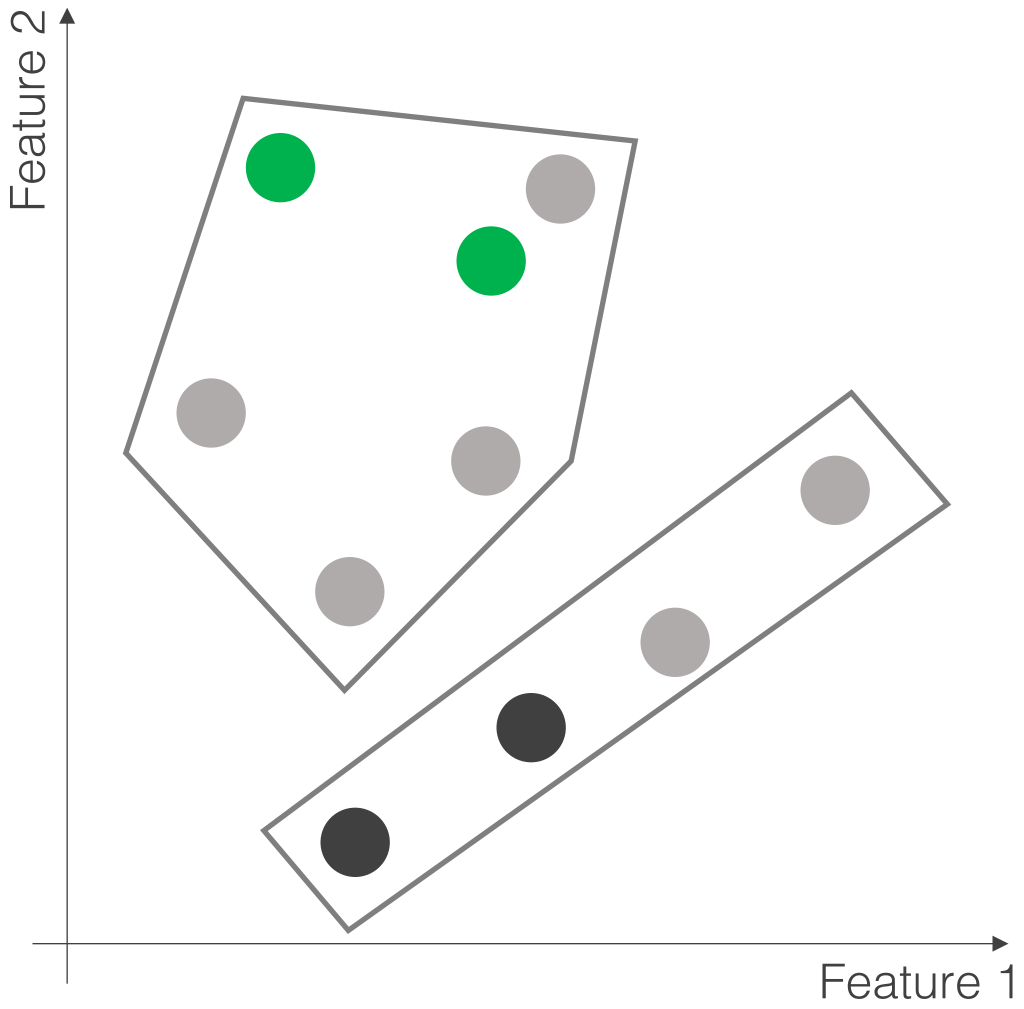


- Have a mix of labeled and unlabeled data
- Want to make predictions from a supervised learning model, $\hat{f}(\boldsymbol{x})$
- Use BOTH the labeled AND unlabeled data for model training

Semi-supervised learning: label propagation

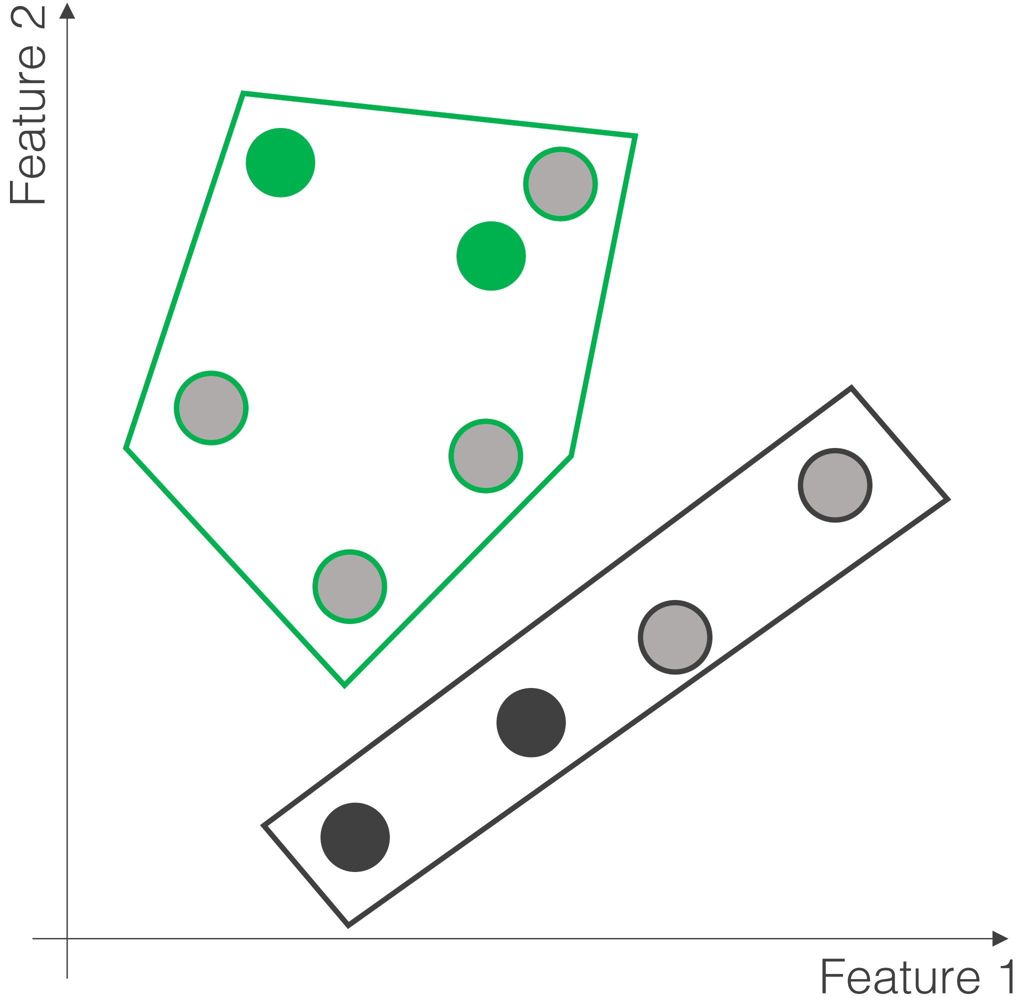


Semi-supervised learning: label propagation



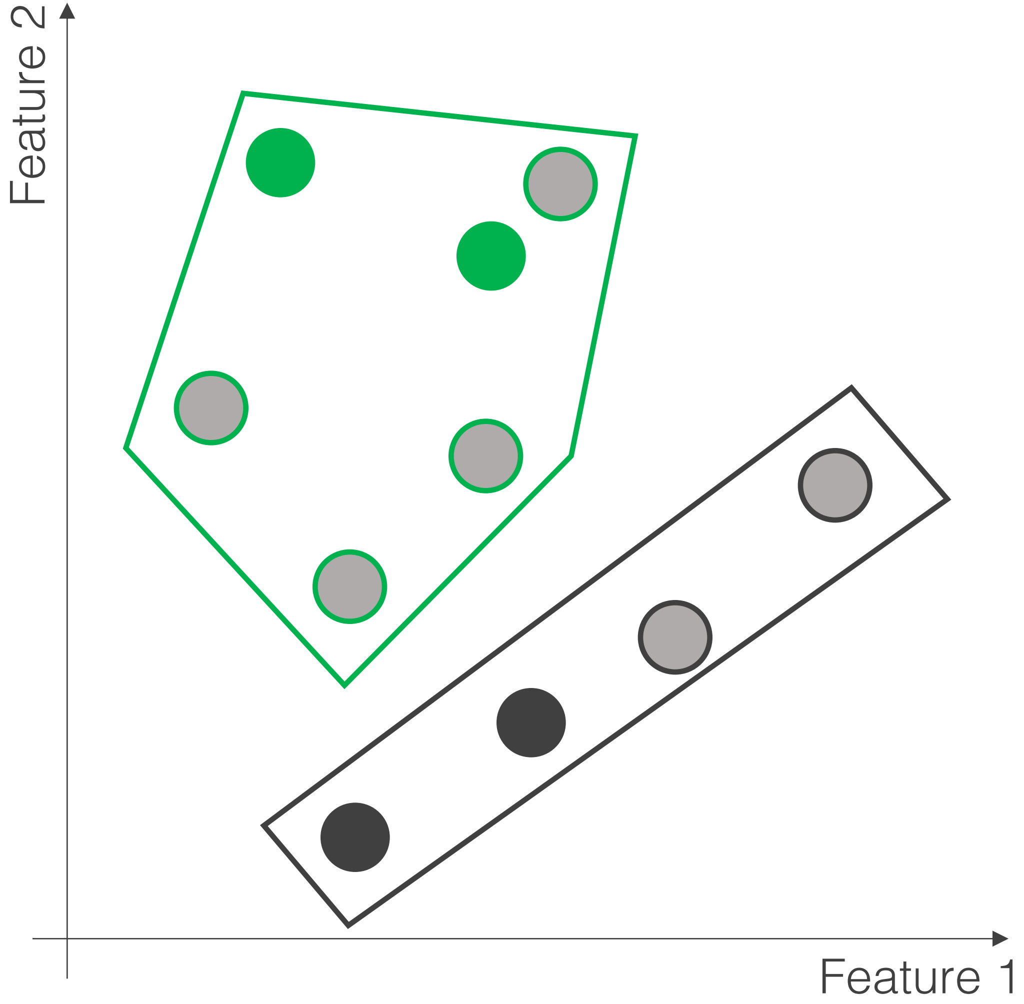
1. Cluster the data such that each cluster has at most one class of labeled data

Semi-supervised learning: label propagation



1. Cluster the data such that each cluster has at most one class of labeled data
2. Assign each sample in each cluster to the corresponding class

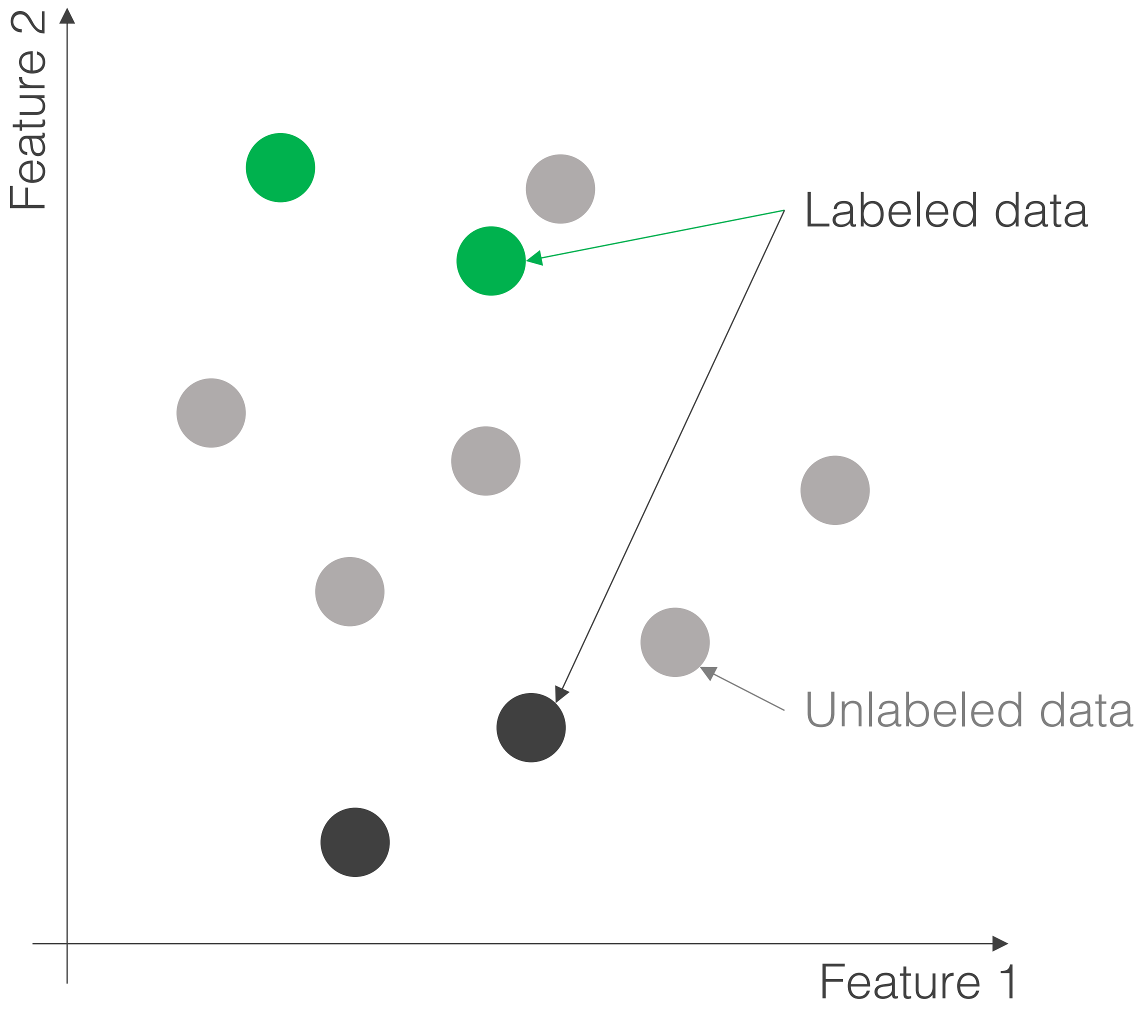
Semi-supervised learning: label propagation



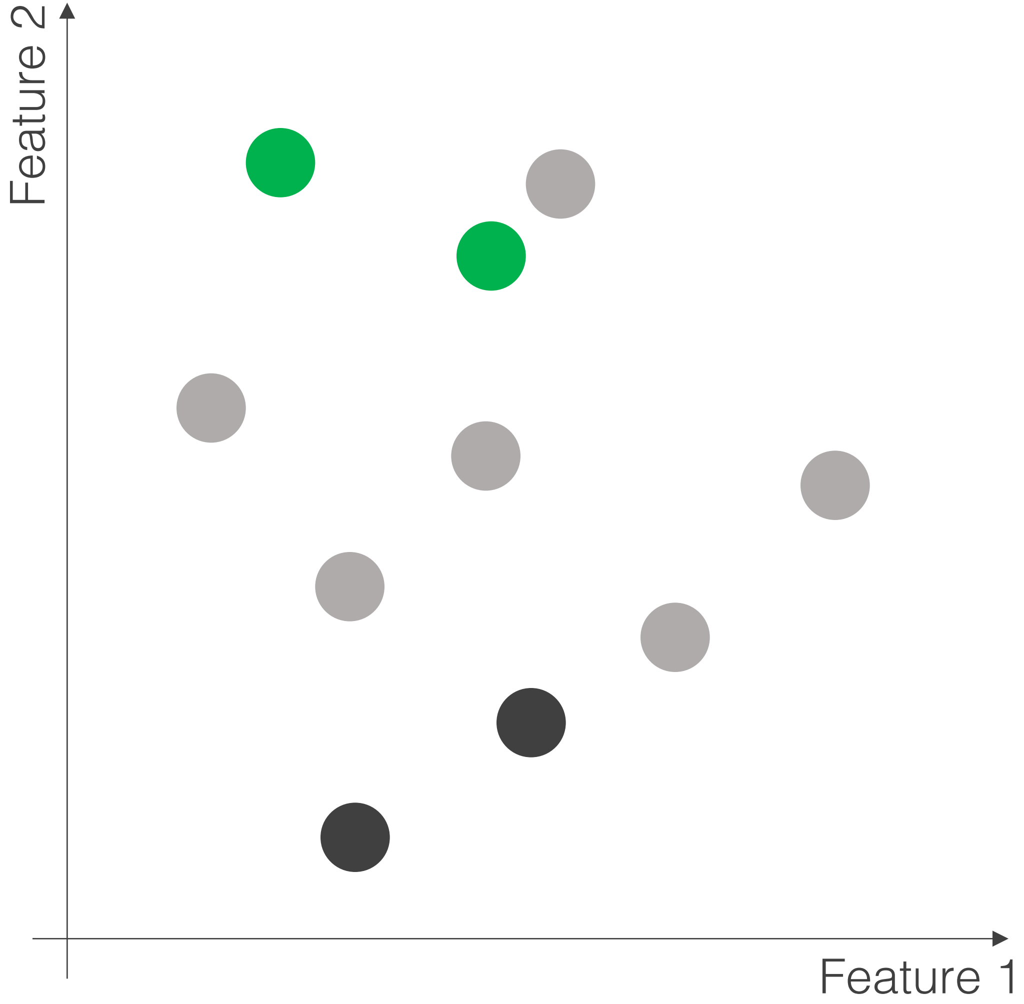
1. Cluster the data such that each cluster has at most one class of labeled data
2. Assign each sample in each cluster to the corresponding class
3. Train a supervised model, $\hat{f}(\boldsymbol{x})$, on the labeled data plus the pseudo-labeled data

- The method of defining clusters / measuring similarity may vary
- Assumes that "similar" points in feature space have similar labels or that clusters share labels

Semi-supervised learning: self-training

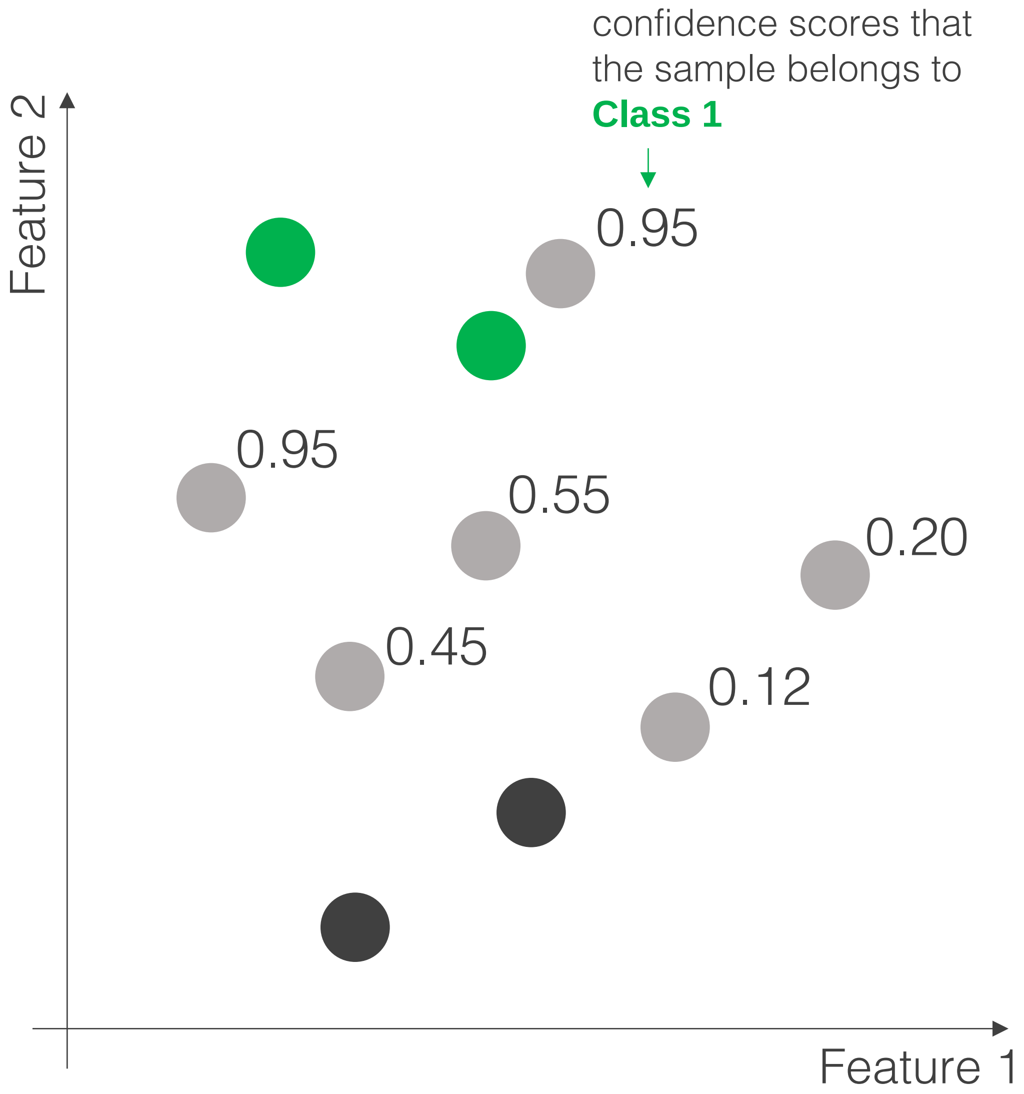


Semi-supervised learning: self-training



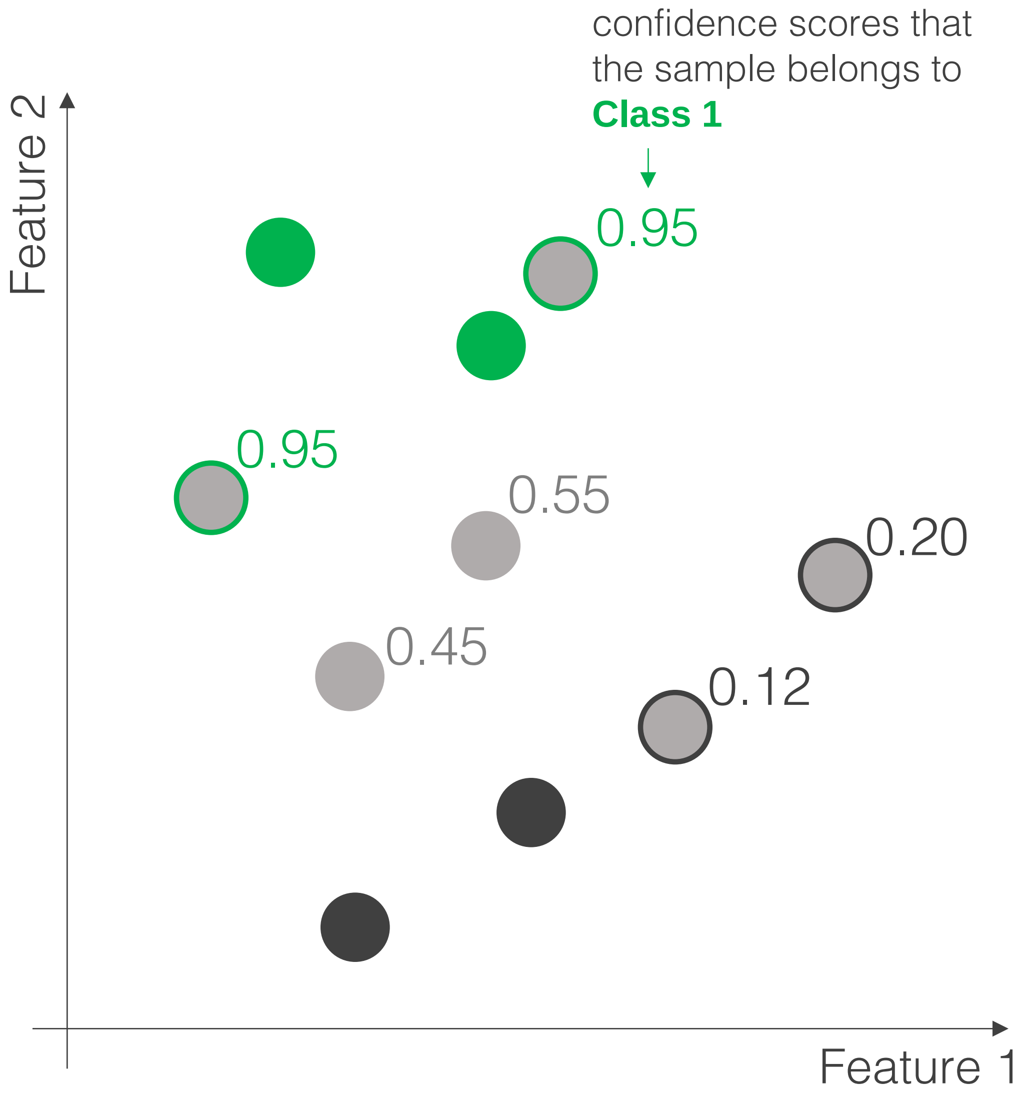
1 Train a supervised model on the labeled data, $\hat{f}(\boldsymbol{x})$

Semi-supervised learning: self-training



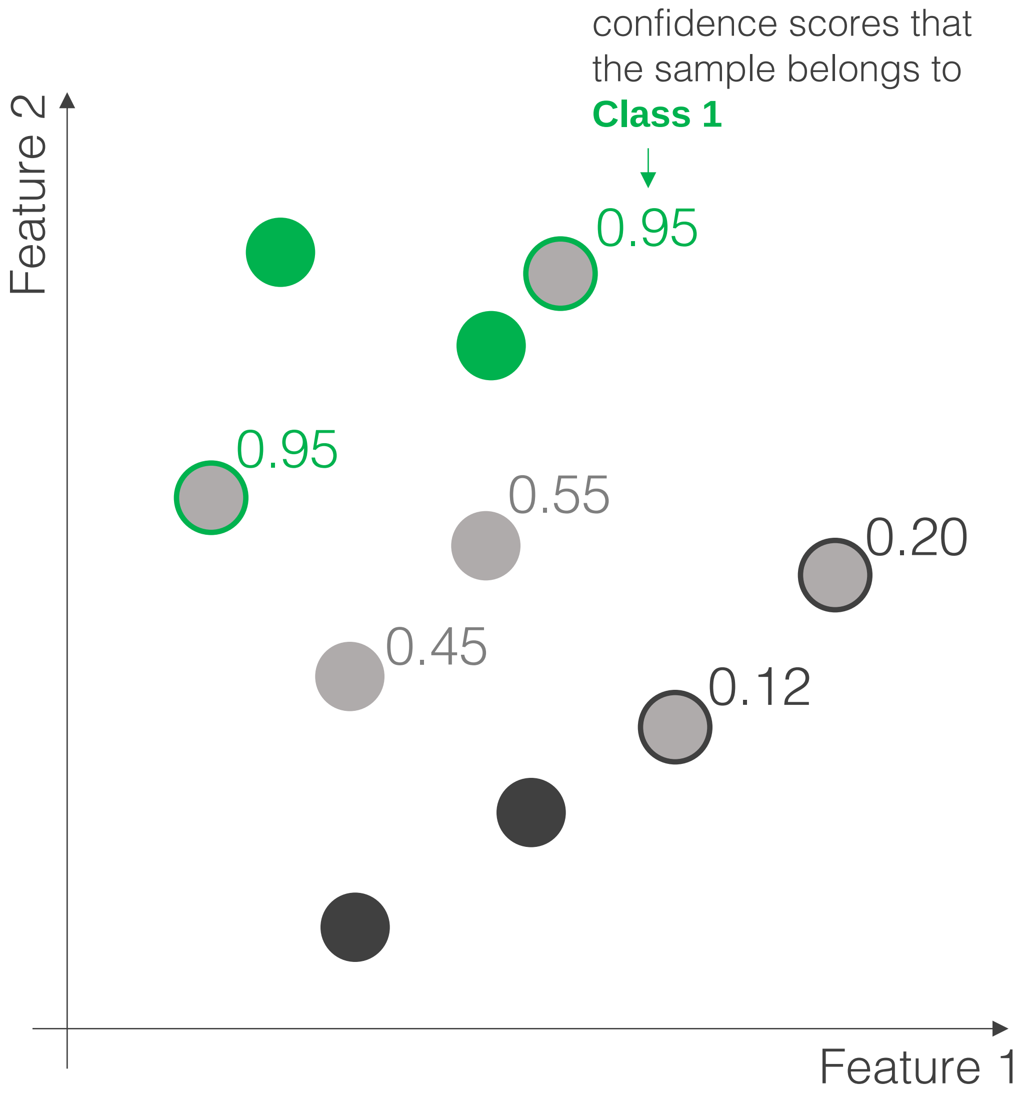
1. Train a supervised model on the labeled data, $\hat{f}(\boldsymbol{x})$
2. Make predictions on the unlabeled data using $\hat{f}(\boldsymbol{x})$

Semi-supervised learning: self-training



1. Train a supervised model on the labeled data, $\hat{f}(\boldsymbol{x})$
2. Make predictions on the unlabeled data using $\hat{f}(\boldsymbol{x})$
3. Use the predictions to assign pseudo-labels to the samples for which the prediction is most confident

Semi-supervised learning: self-training



1. Train a supervised model on the labeled data, $\hat{f}(\boldsymbol{x})$
2. Make predictions on the unlabeled data using $\hat{f}(\boldsymbol{x})$
3. Use the predictions to assign pseudo-labels to the samples for which the prediction is most confident
4. Retrain the model, $\hat{f}(\boldsymbol{x})$, using BOTH the labels and pseudo-labels

Refresher: Loss / Cost functions

$$L(\boldsymbol{X},\boldsymbol{y},\boldsymbol{w}) = E(\boldsymbol{X},\boldsymbol{y}) \quad + \quad \lambda R(\boldsymbol{w})$$

Regression
(mean squared error)

$$L(\boldsymbol{X},\boldsymbol{y},\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{f}(\boldsymbol{x}_i)\right)^2 \quad + \quad \lambda \sum_{j=1}^{p} w_j^2$$

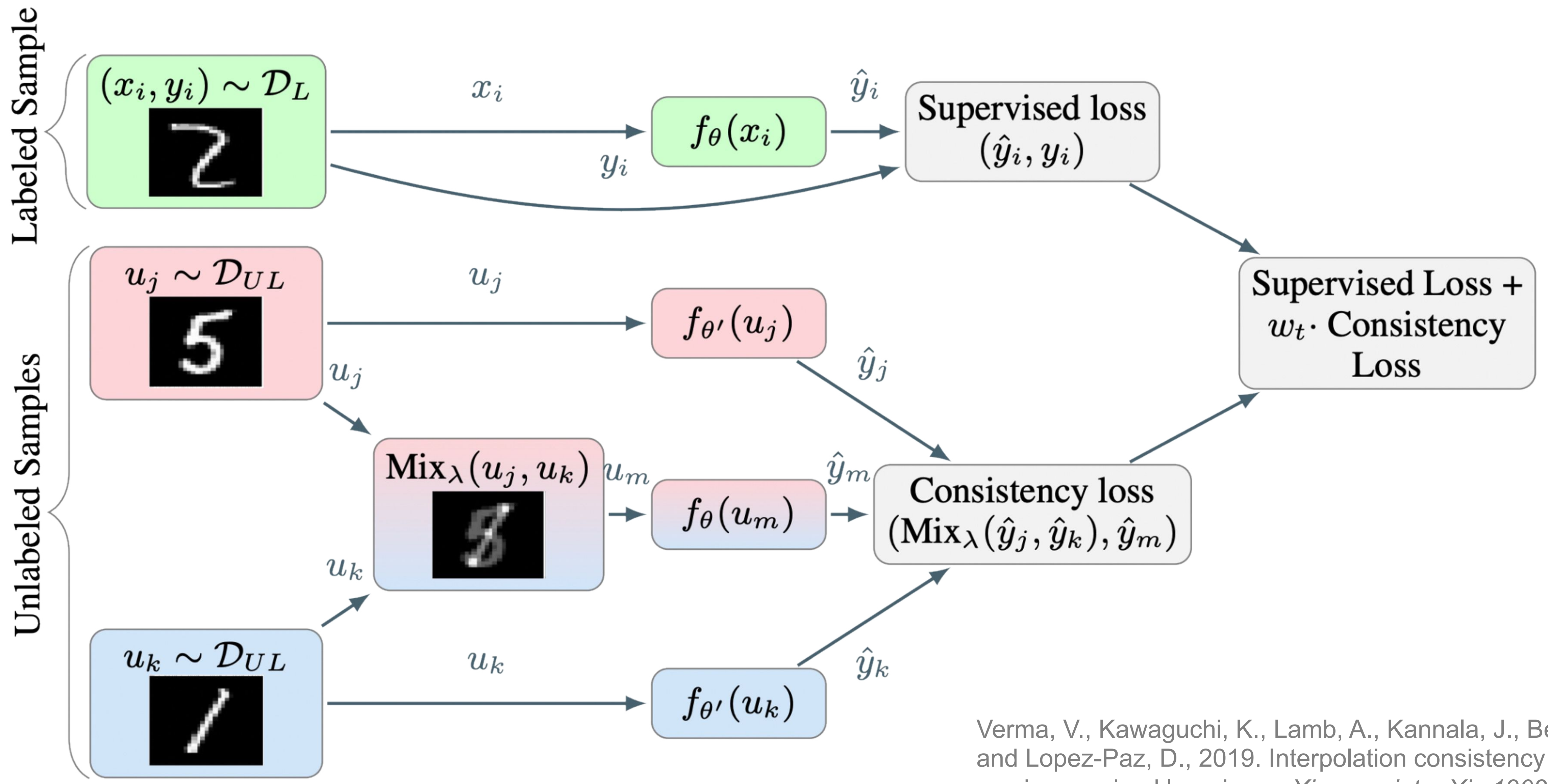
Mean square error

L_2 regularization penalty can be added to either

Classification
(average binary cross entropy)

$$L(\boldsymbol{X},\boldsymbol{y},\boldsymbol{w}) = -\frac{1}{N}\left[y_i \log\left(\hat{f}(\boldsymbol{x}_i)\right) + (1-y_i)\log\left(1-\hat{f}(\boldsymbol{x}_i)\right)\right] + \lambda \sum_{j=1}^{p} w_j^2$$

Semi-supervised learning: consistency regularization



Verma, V., Kawaguchi, K., Lamb, A., Kannala, J., Bengio, Y. and Lopez-Paz, D., 2019. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825*.

Semi-supervised learning summary

Allows the use of BOTH labeled and unlabeled data

Reduces the cost of labeling processes

Requires making some strong assumptions about the data, e.g.:

- Points that are close to each other are more likely to share a label
- Points exist in clusters and are likely to share the same label within a cluster

Does not always improve performance

Special Topics

Semi-supervised learning

Self-supervised learning

Recommender systems

Other Practical Considerations

Self-supervised learning

The data do not come with labels – **we “make” our own labels**

The approaches used are **supervised** in nature

These methods can then be used for supervised learning problems through **transfer learning**

Recall Autoencoders

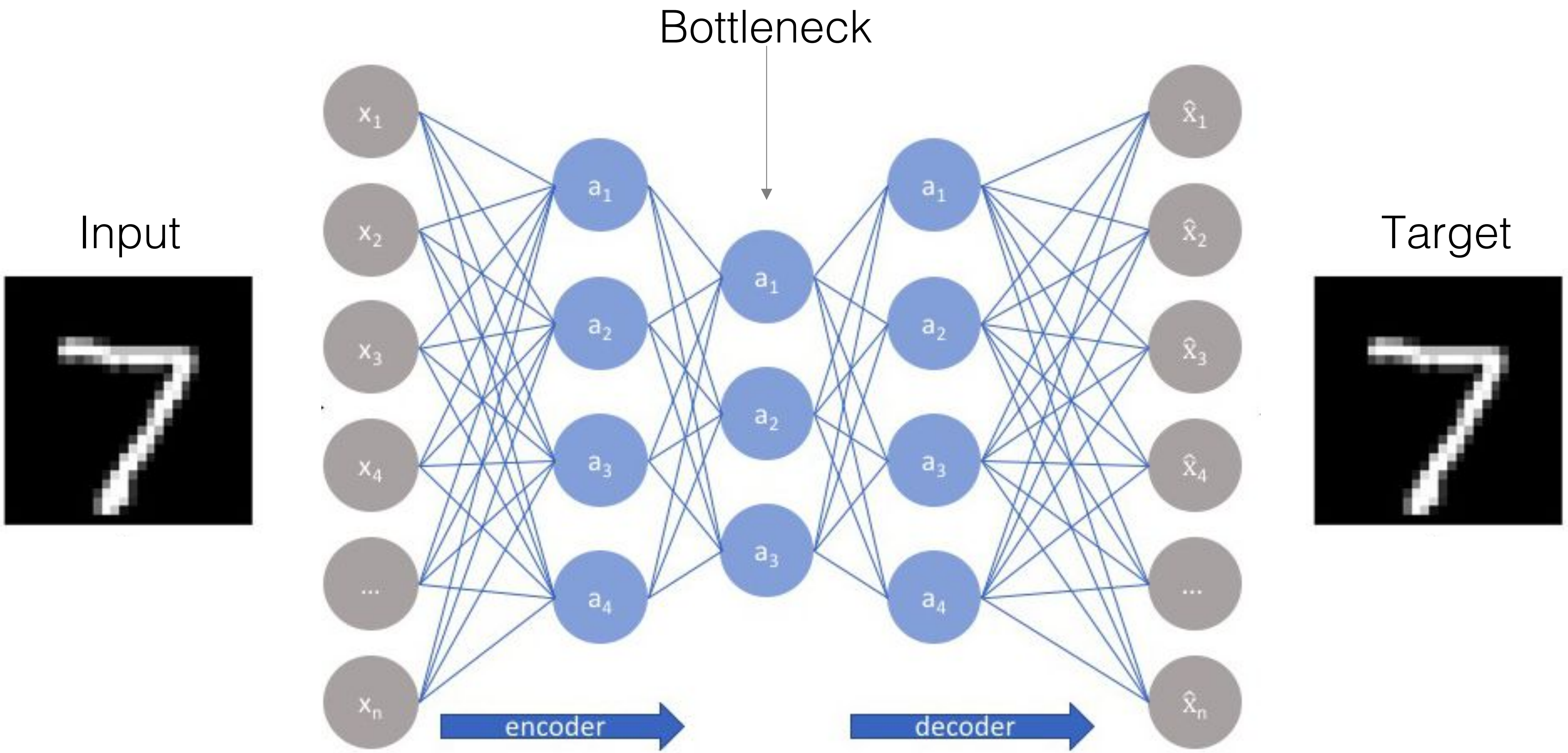
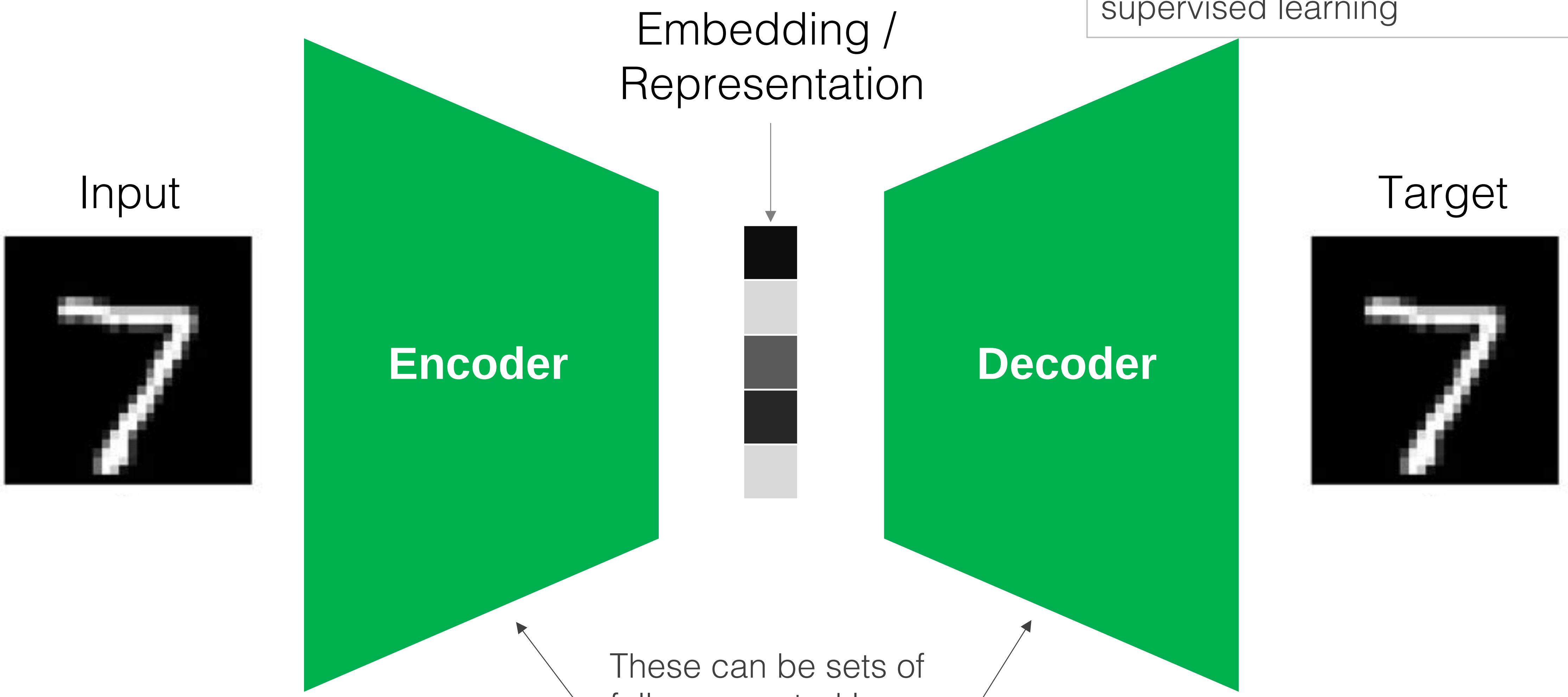


Image from: https://www.jeremyjordan.me/autoencoders/

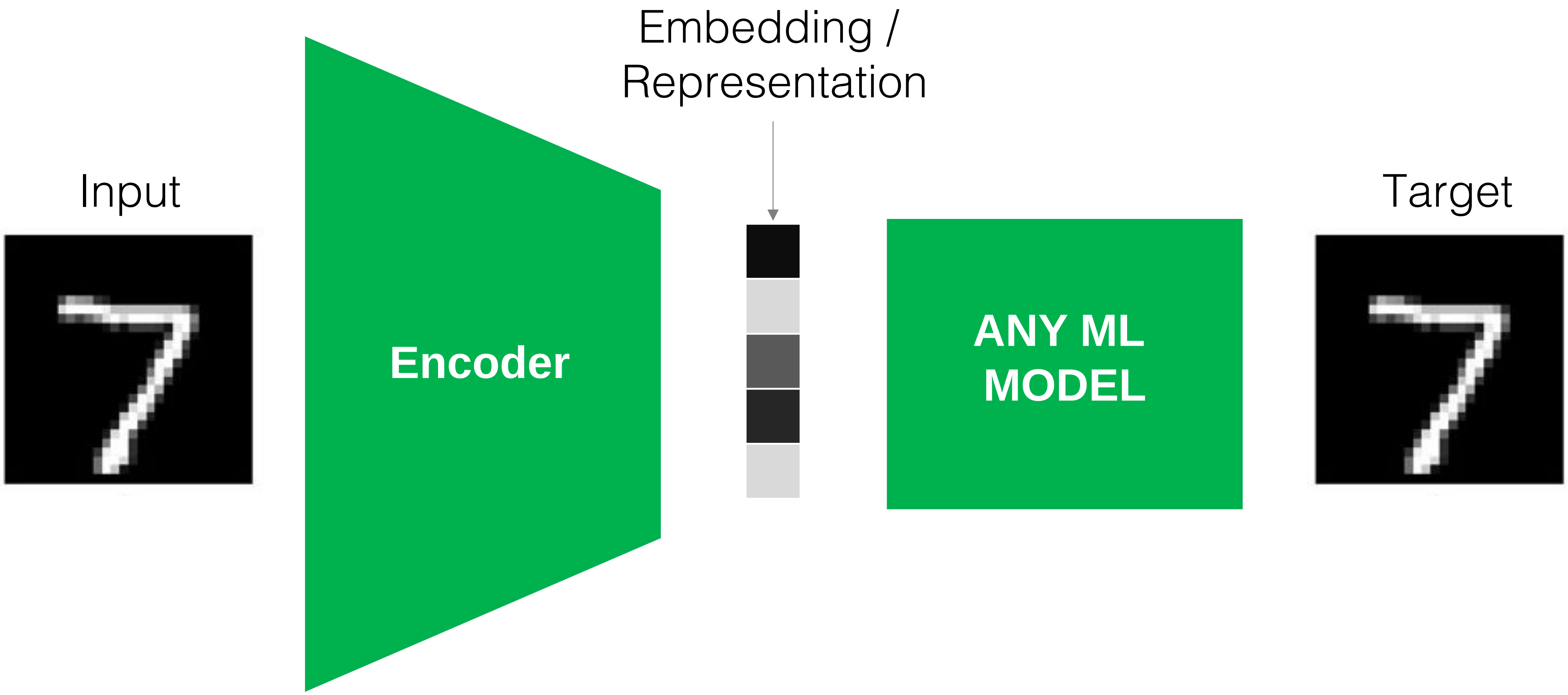
Recall Autoencoders

Our goal is often to develop a good **encoder** that represents our features well: this is a core insight of self-supervised learning

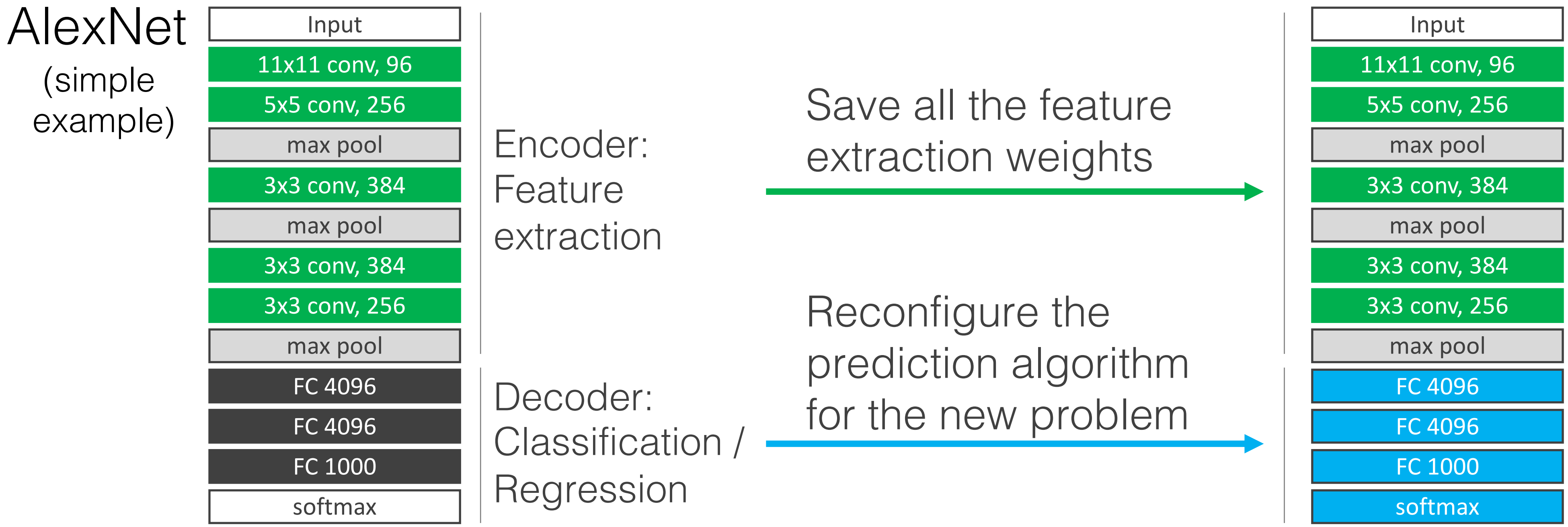


These can be sets of fully-connected layers or convolutional layers, etc.

Recall Autoencoders



Transfer learned feature representations



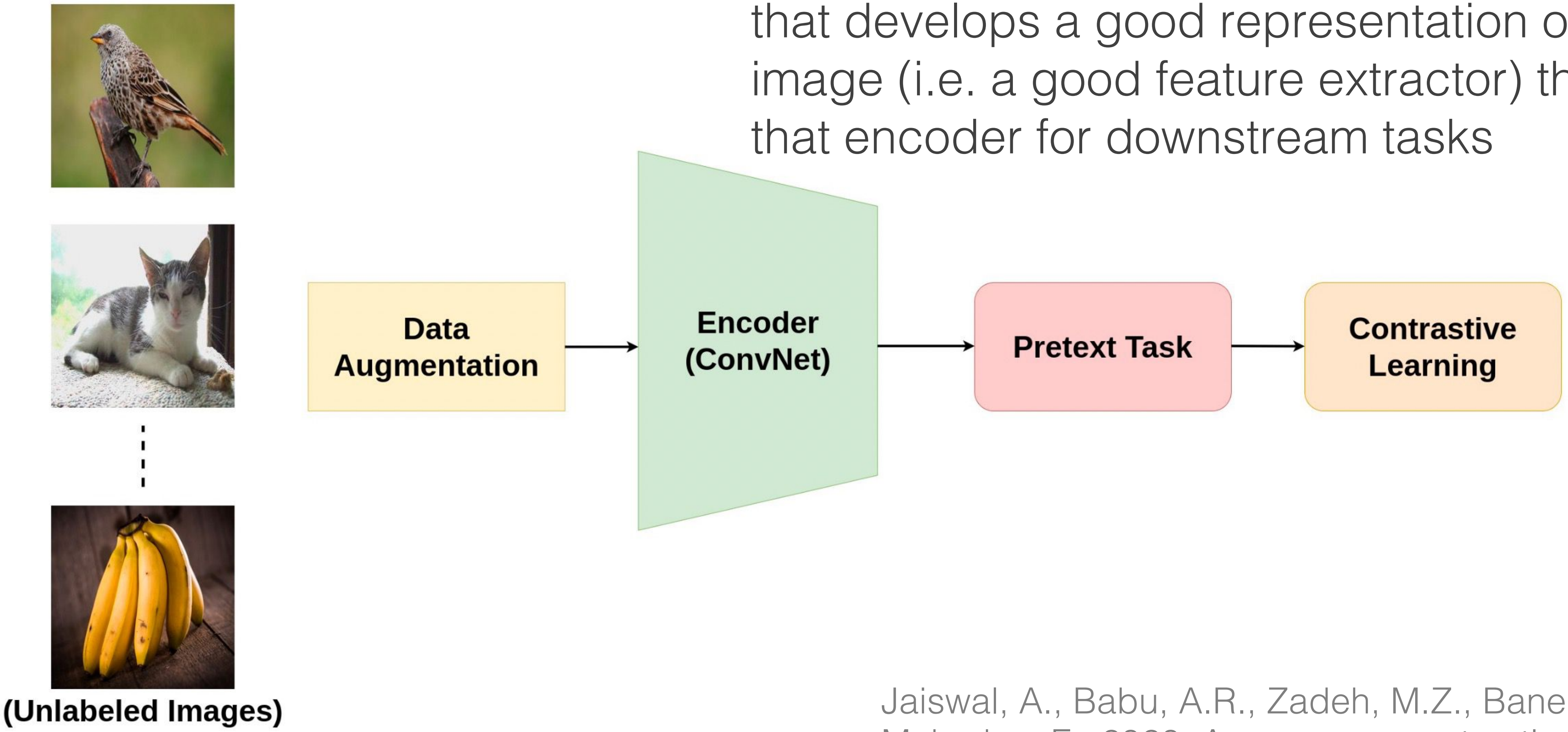
Train a model on dataset A

Can either use features as-is OR fine-tune a model on dataset B

(fine-tune = retrain model a little with saved weights)

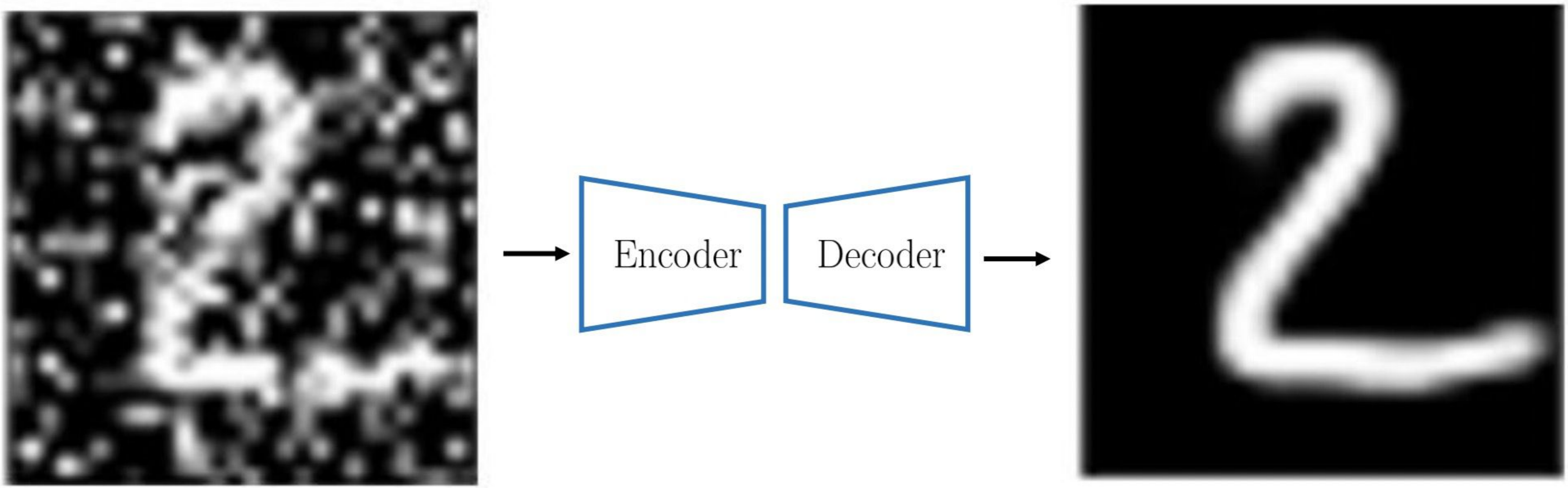
Self-supervised learning: contrastive learning

Concept: Train an encoder on a "pretext" task that develops a good representation of the image (i.e. a good feature extractor) then use that encoder for downstream tasks



Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D. and Makedon, F., 2020. A survey on contrastive self-supervised learning. Technologies, 9(1), p.2.

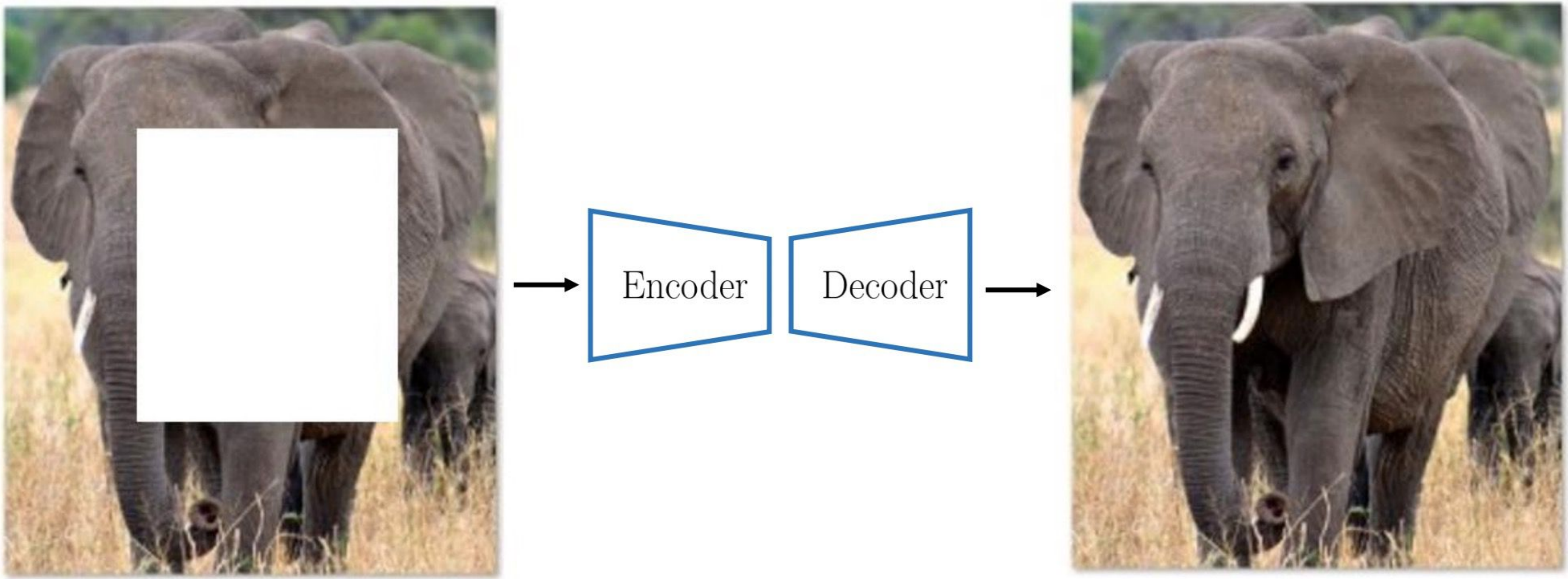
Pretext task example: denoising



A pretext task creates labeled data from unlabeled data

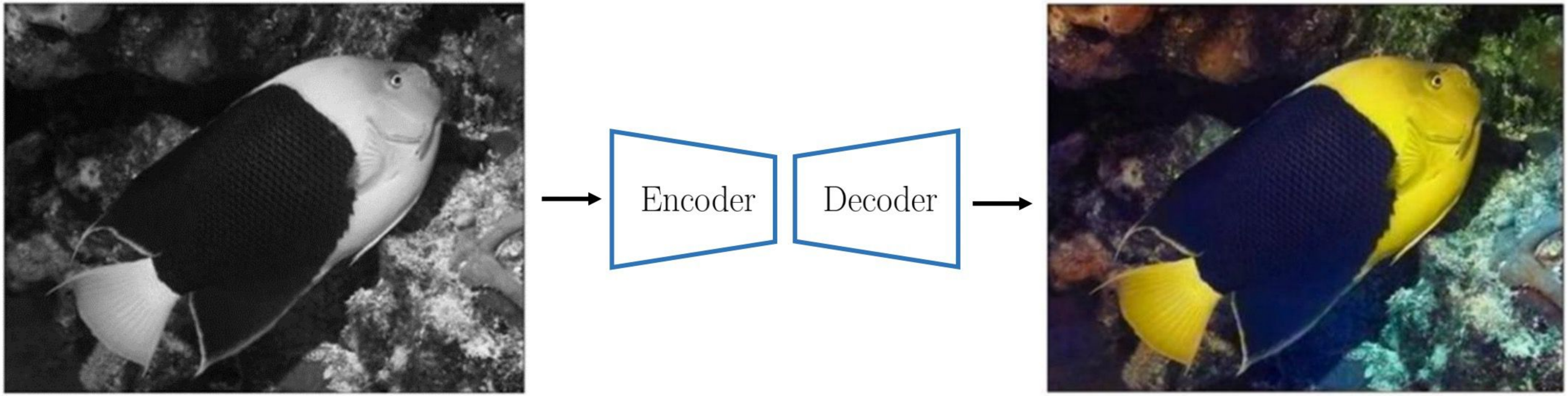
Spyros Gidaris and Andrei Bursuc. 2021. Teacher-student feature prediction approaches. CVPR 2021 Tutorial on Leave Those Nets Alone: Advances in Self-Supervised Learning (link).

Pretext task example: image inpainting



Spyros Gidaris and Andrei Bursuc. 2021. Teacher-student feature prediction approaches.
CVPR 2021 Tutorial on Leave Those Nets Alone: Advances in Self-Supervised Learning (link).

Pretext task example: colorization

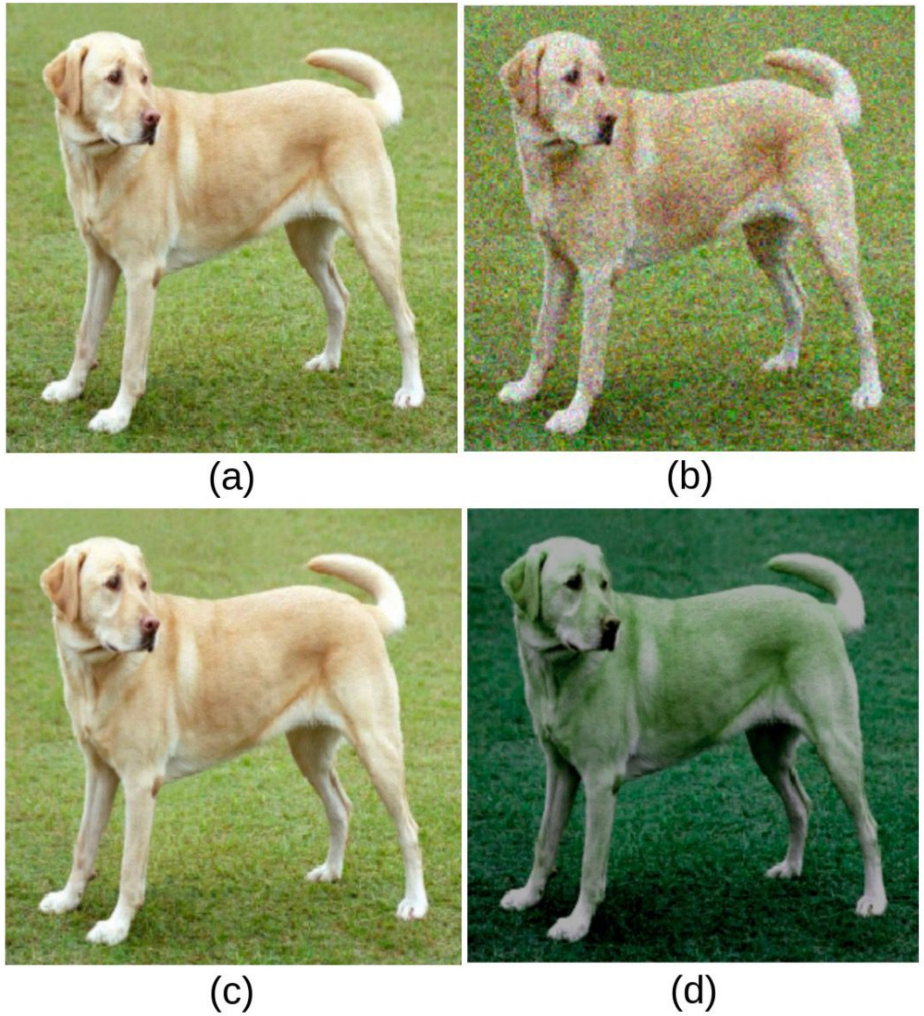


Spyros Gidaris and Andrei Bursuc. 2021. Teacher-student feature prediction approaches.
CVPR 2021 Tutorial on Leave Those Nets Alone: Advances in Self-Supervised Learning (link).

Augmentations that may be used as pretext tasks for images

Color transformations

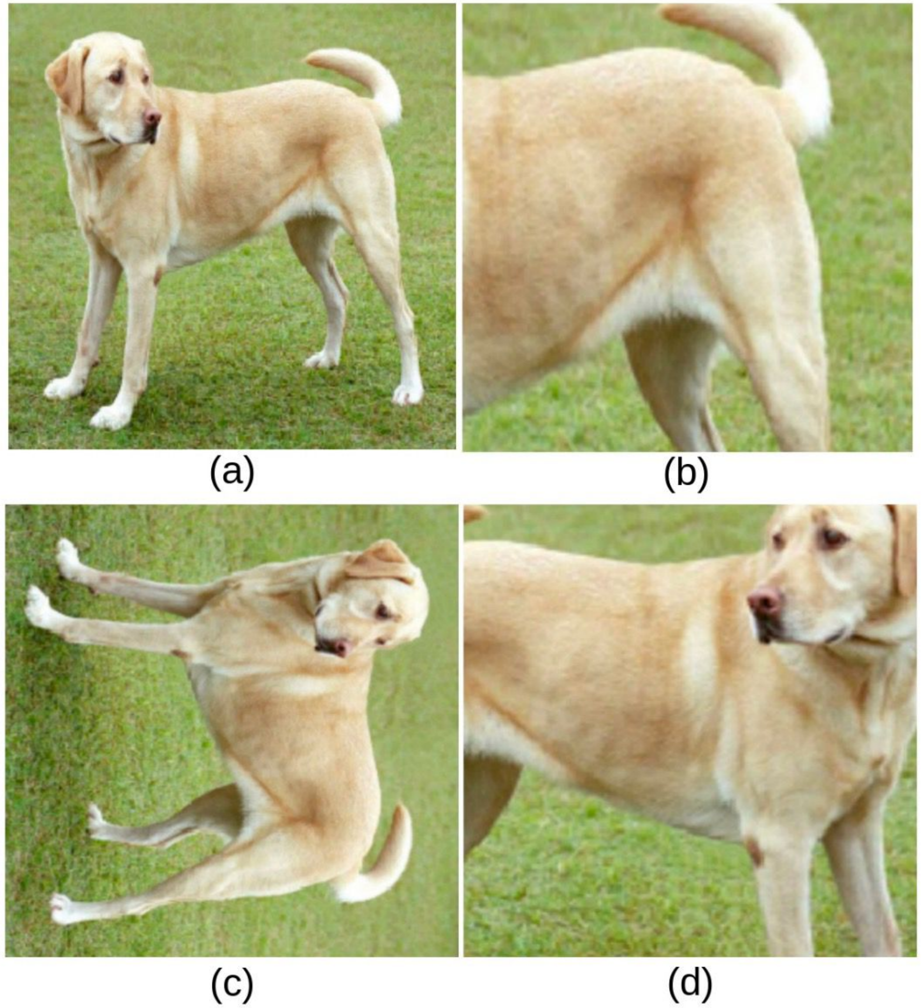
Original | Noise added



Blurring | Color distortion

Geometric transformations

Original | Crop



Rotation | Crop, resize, flip

Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D. and Makedon, F., 2020. A survey on contrastive self-supervised learning. Technologies, 9(1), p.2

NLP Pretext task examples

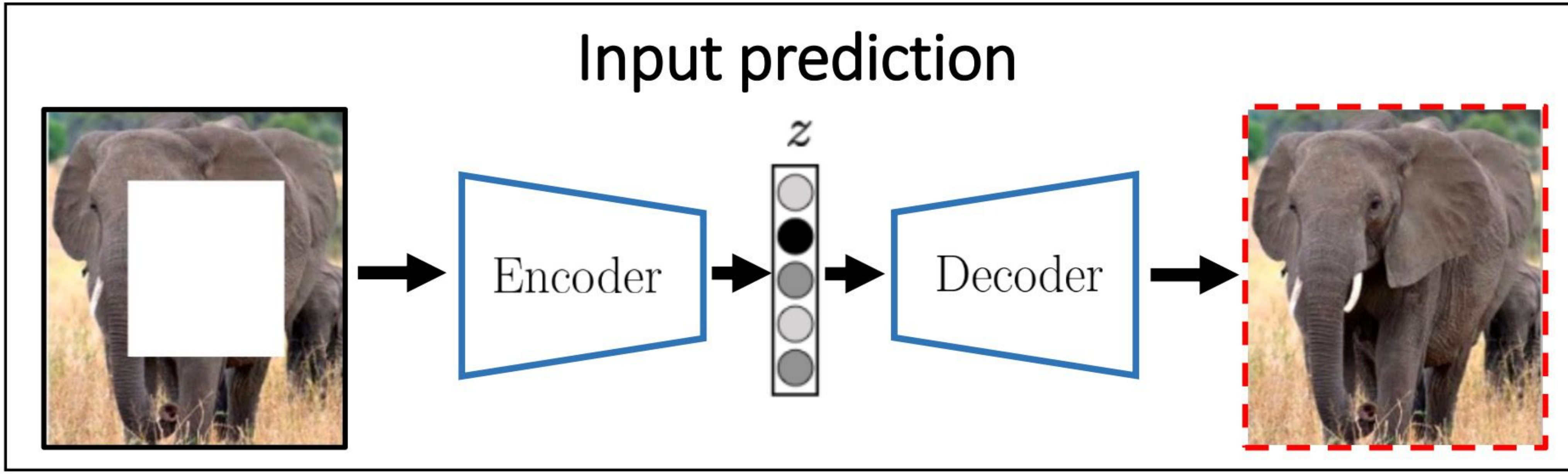
Center word prediction

Neighbor prediction

Masked word prediction

Other examples include: sentence order prediction, sentence shuffling

Images from Amit Chaudhary: https://amitness.com/2020/05/self-supervised-learning-nlp/

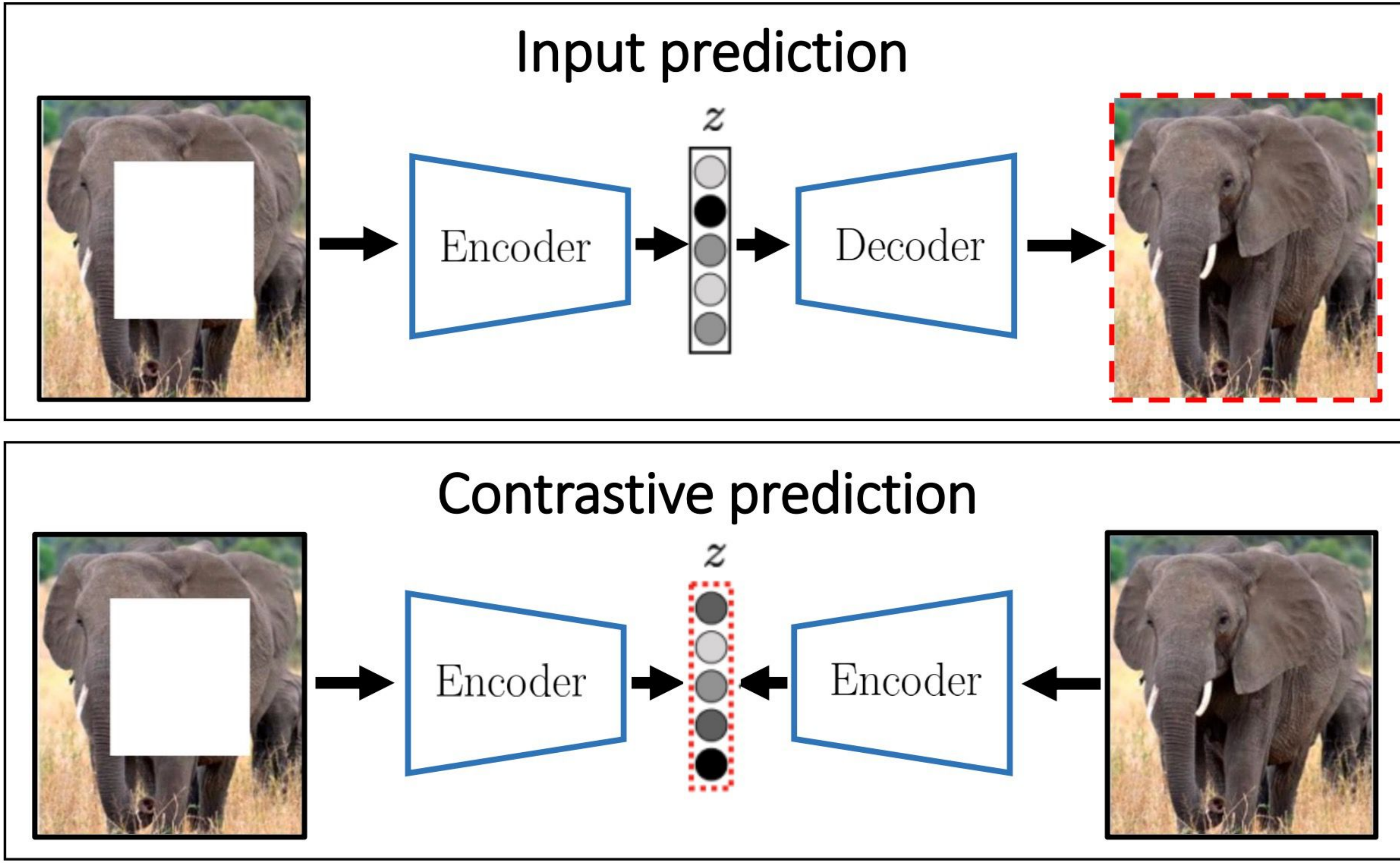


Problem: this approach focuses on a lot of "useless" work: specific details of color, texture, and shapes

We want to have the algorithm represent the "concept" of the elephant and tell that the two images are the same

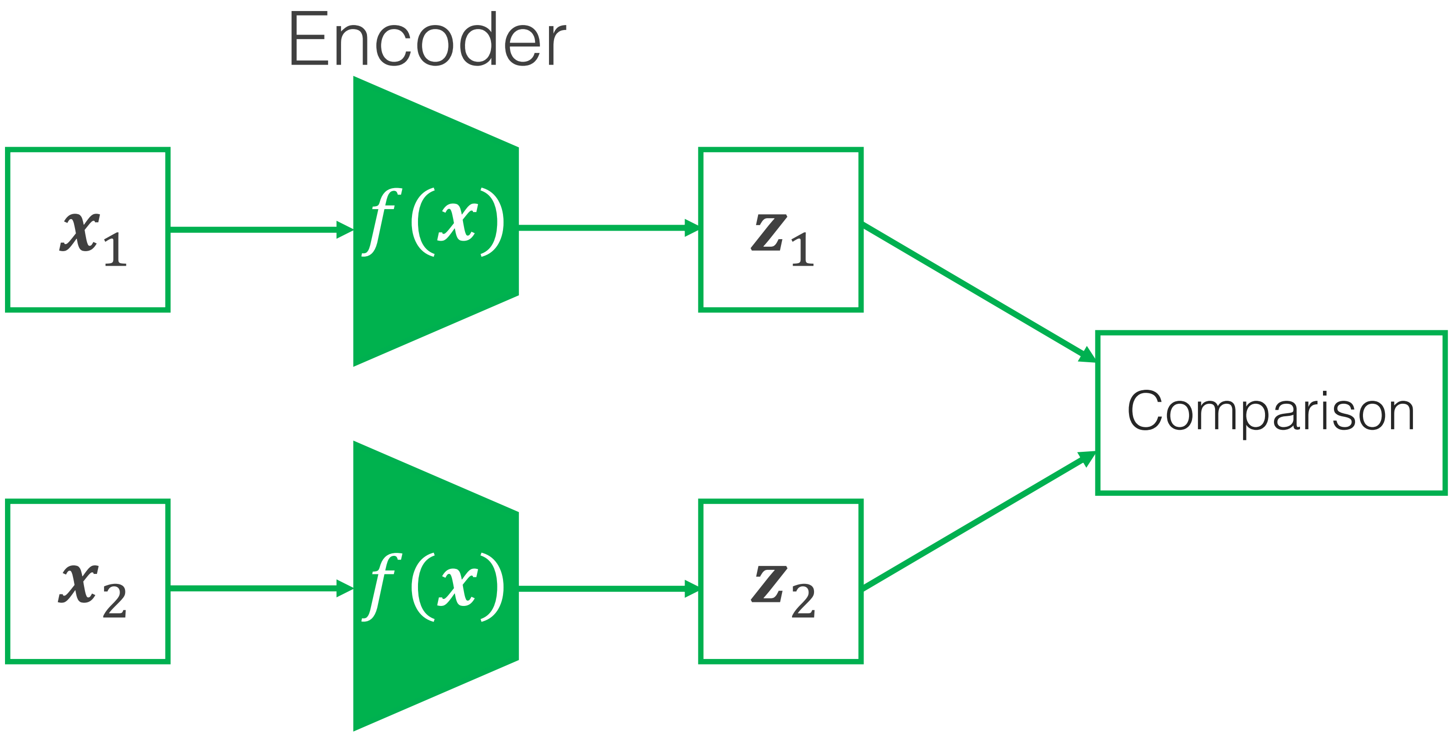
Spyros Gidaris and Andrei Bursuc. 2021. Teacher-student feature prediction approaches. CVPR 2021 Tutorial on Leave Those Nets Alone: Advances in Self-Supervised Learning ([link]()).

Contrastive learning adjusts the loss / cost function to train the representation z to be similar for both images



Spyros Gidaris and Andrei Bursuc. 2021. Teacher-student feature prediction approaches. CVPR 2021 Tutorial on Leave Those Nets Alone: Advances in Self-Supervised Learning (link).

Self-supervised contrastive learning



Minimize the representation distance between the “similar” samples

Maximize the representation distance between the “similar” samples

Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D. and Makedon, F., 2020. A survey on contrastive self-supervised learning. Technologies, 9(1), p.2.

Triplet loss

Anchor

Neighbor

Distant

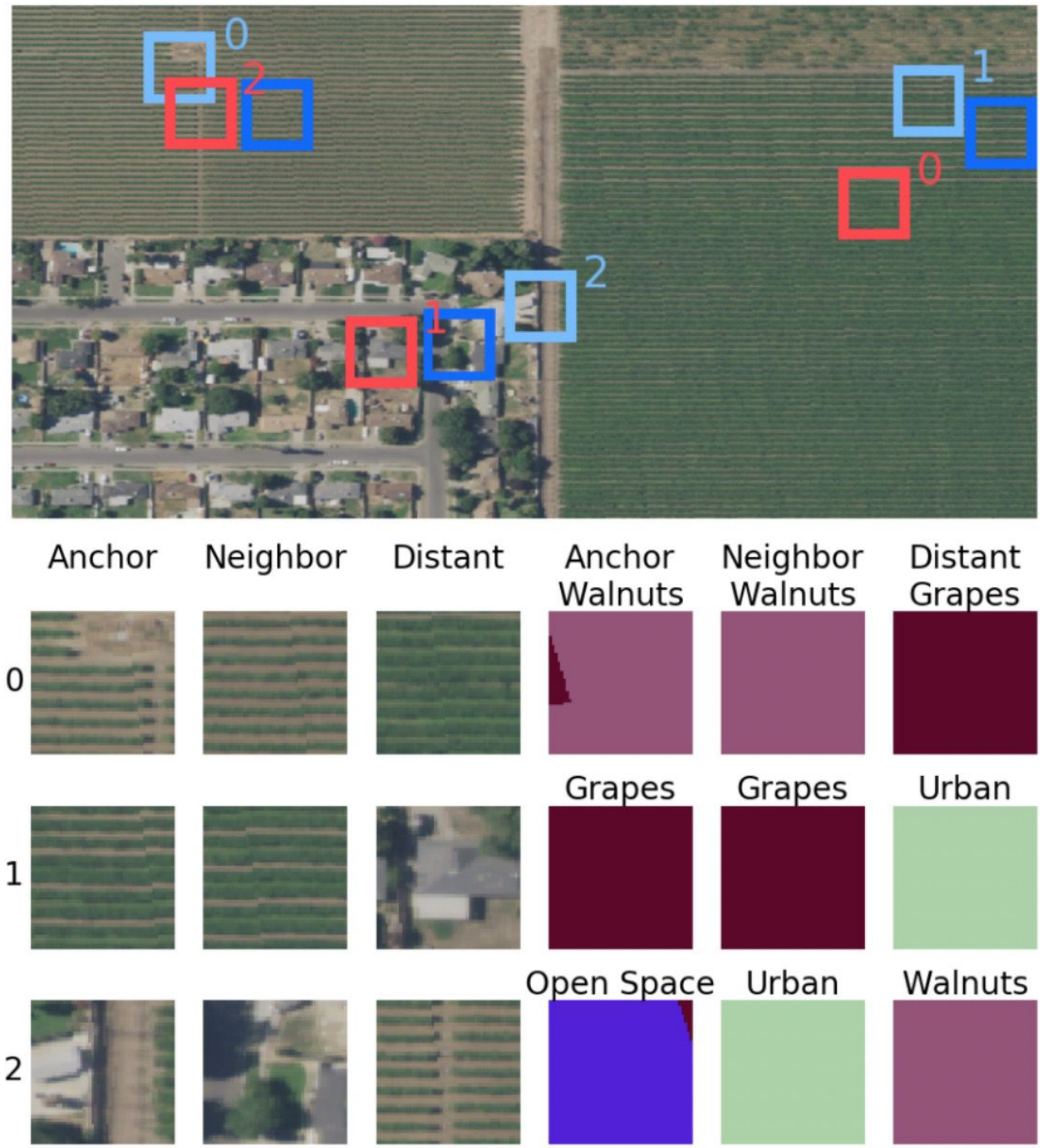
$$L(\boldsymbol{x}_a, \boldsymbol{x}_n, \boldsymbol{x}_d) =$$

$$\left\|\hat{f}(\boldsymbol{x}_a) - \hat{f}(\boldsymbol{x}_n)\right\|_2$$

Minimize the distance of the neighbors

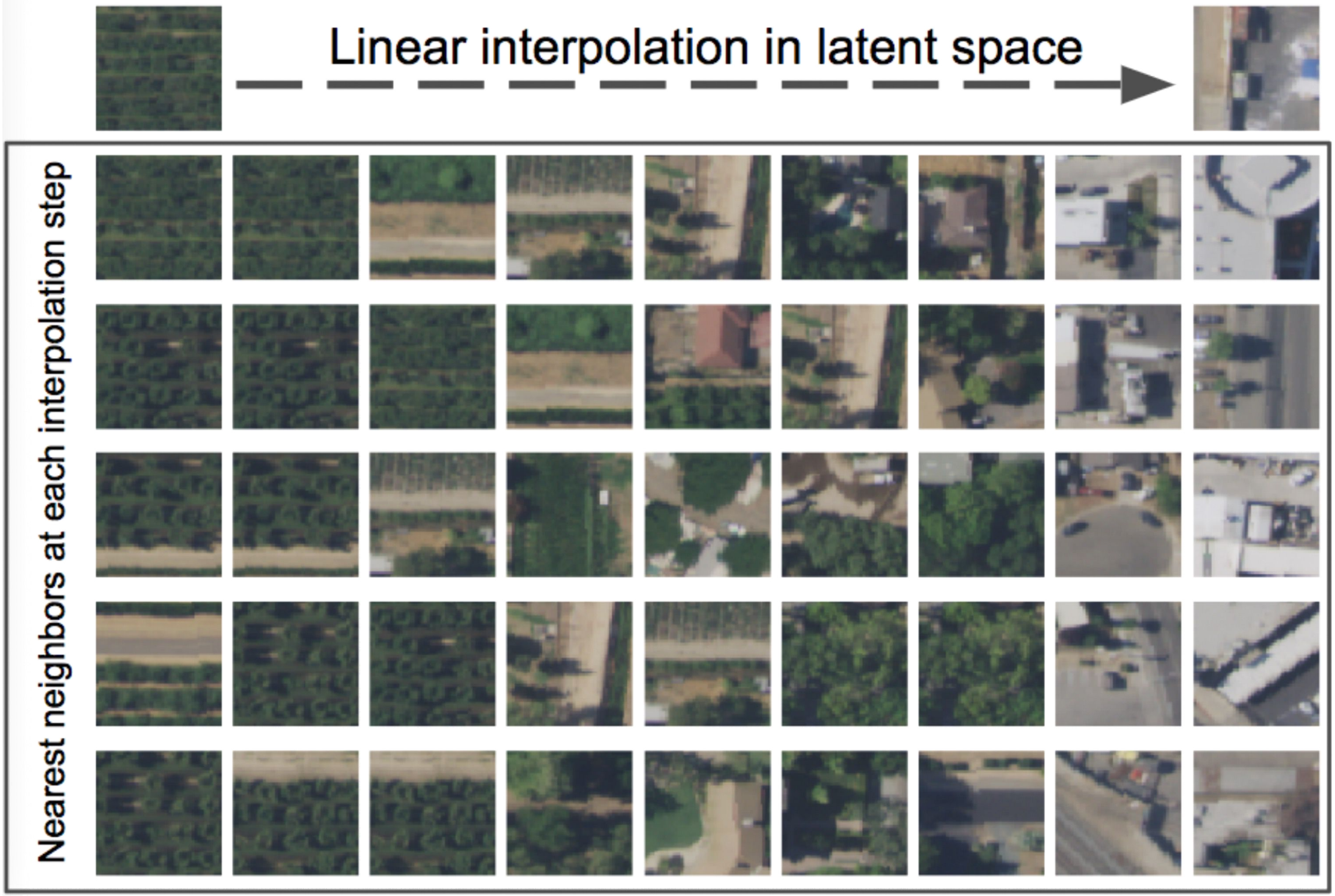
$$-\left\|\hat{f}(\boldsymbol{x}_a) - \hat{f}(\boldsymbol{x}_d)\right\|_2$$

Maximize the distance of the "distant" images



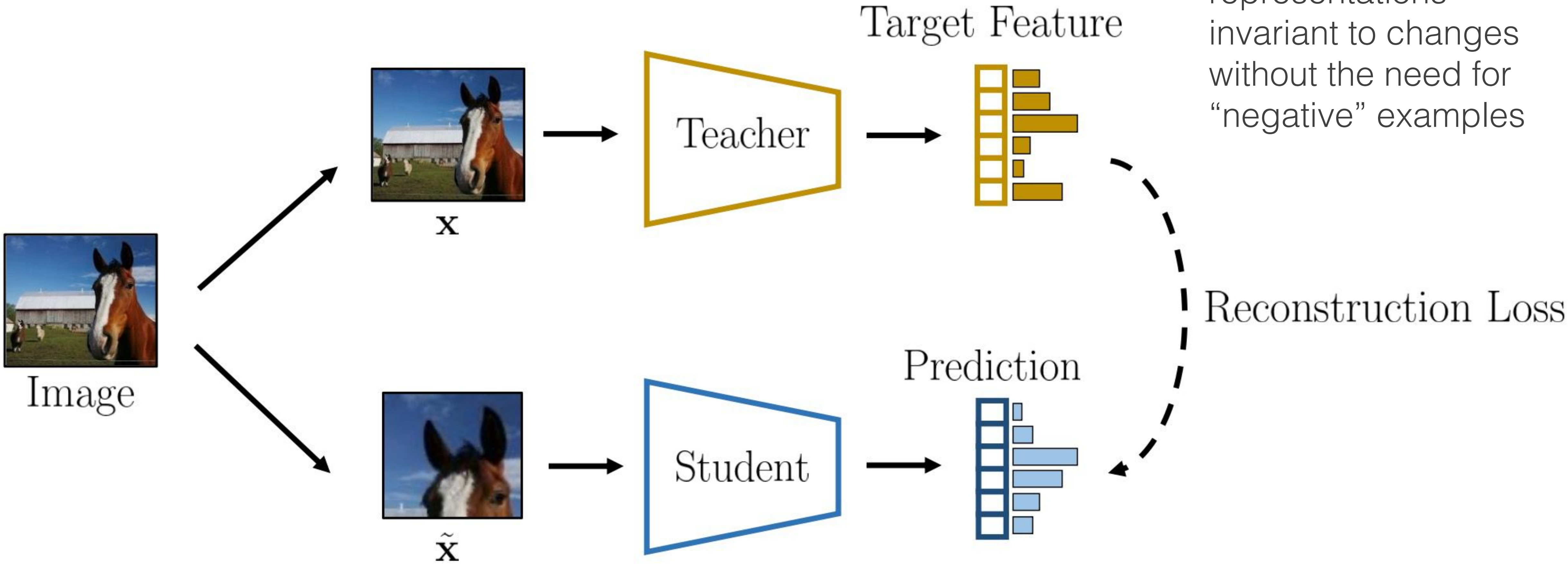
Jean, N., Wang, S., Samar, A., Azzari, G., Lobell, D. and Ermon, S., 2019, July. Tile2vec: Unsupervised representation learning for spatially distributed data. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 3967-3974).

Triplet loss Results



Jean, N., Wang, S., Samar, A., Azzari, G., Lobell, D. and Ermon, S., 2019, July. Tile2vec: Unsupervised representation learning for spatially distributed data. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 3967-3974).

Self-supervised contrastive learning

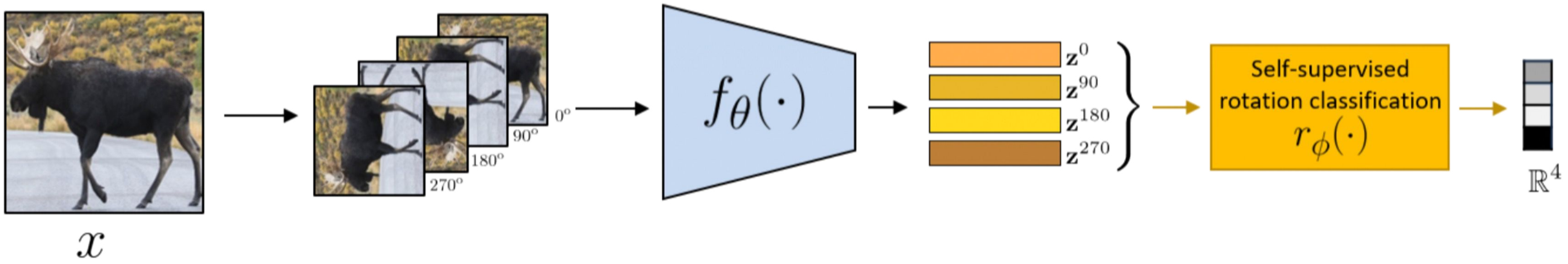


Develop similar feature representations invariant to changes without the need for “negative” examples

Spyros Gidaris and Andrei Bursuc. 2021. Teacher-student feature prediction approaches. CVPR 2021 Tutorial on Leave Those Nets Alone: Advances in Self-Supervised Learning (link).

Self-supervised learning → downstream tasks

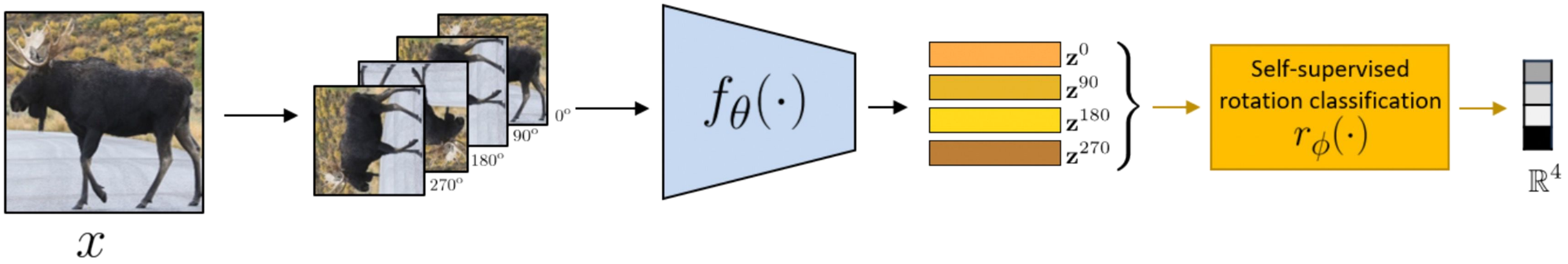
Stage 1: Train network on pretext task (without human labels)



Andrei Bursuc and Spyros Gidaris. 2021. Introduction to Self-supervised Learning. CVPR 2021 Tutorial on Leave Those Nets Alone: Advances in Self-Supervised Learning (link).

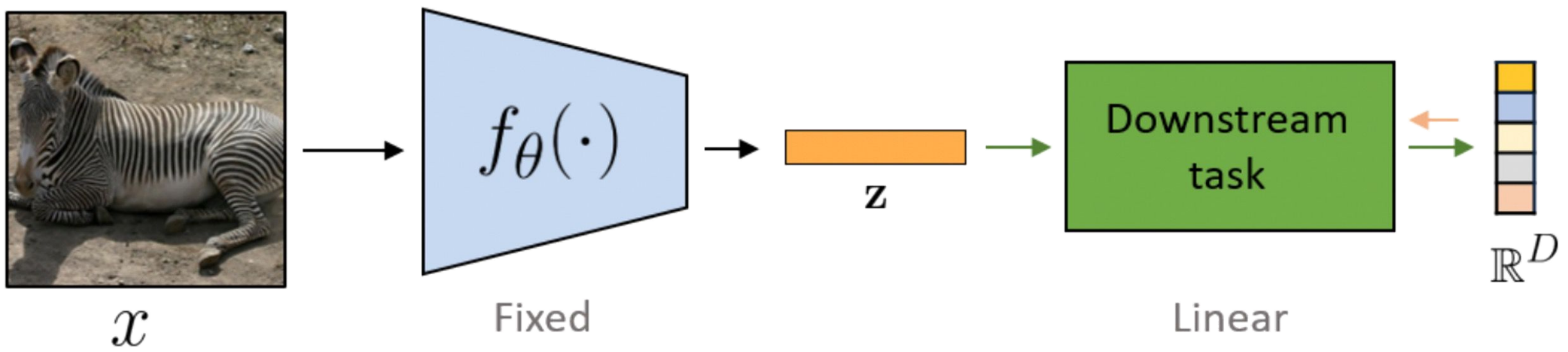
Self-supervised learning → downstream tasks

Stage 1: Train network on pretext task (without human labels)



Stage 2: Train classifier on learned features for new task with fewer labels

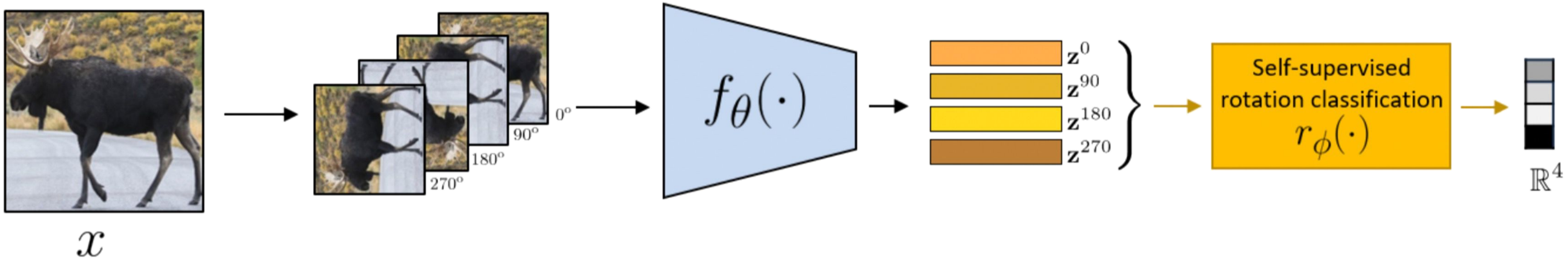
The encoder becomes a pretrained model for downstream tasks through transfer learning



Andrei Bursuc and Spyros Gidaris. 2021. Introduction to Self-supervised Learning. CVPR 2021 Tutorial on Leave Those Nets Alone: Advances in Self-Supervised Learning (link).

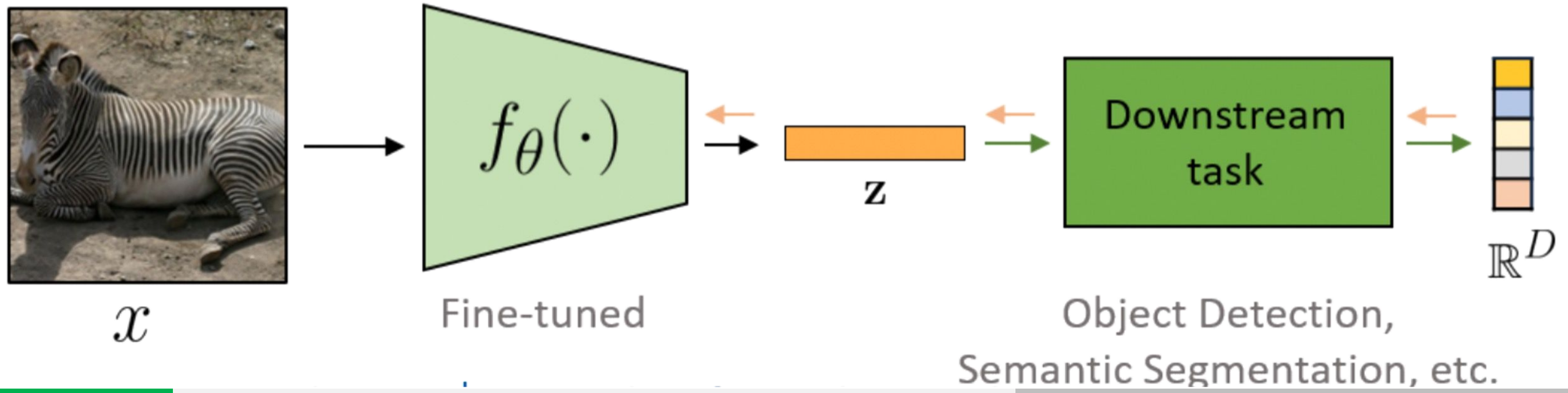
Self-supervised learning → downstream tasks

Stage 1: Train network on pretext task (without human labels)



Stage 2: Fine-tune network for new task with fewer labels

The encoder becomes a pretrained model for downstream tasks through transfer learning



Andrei Bursuc and Spyros Gidaris. 2021. Introduction to Self-supervised Learning. CVPR 2021 Tutorial on Leave Those Nets Alone: Advances in Self-Supervised Learning (link).

Self-supervised learning summary

Comes in many flavors: contrastive, teacher-student, etc.

Has generated exceptional NLP models: BERT, GPT-3, word2vec

No labels required!

Large unlabeled dataset required

Massive computation required!

Special Topics

Semi-supervised learning

Self-supervised learning

Recommender systems

Other Practical Considerations

Recommender Systems: collaborative filtering

User-based nearest neighbor

Predict a user rating based on users similar to them

1. Find users who have liked similar things as Alice in the past
2. Average the ratings of those users to make a prediction for Alice

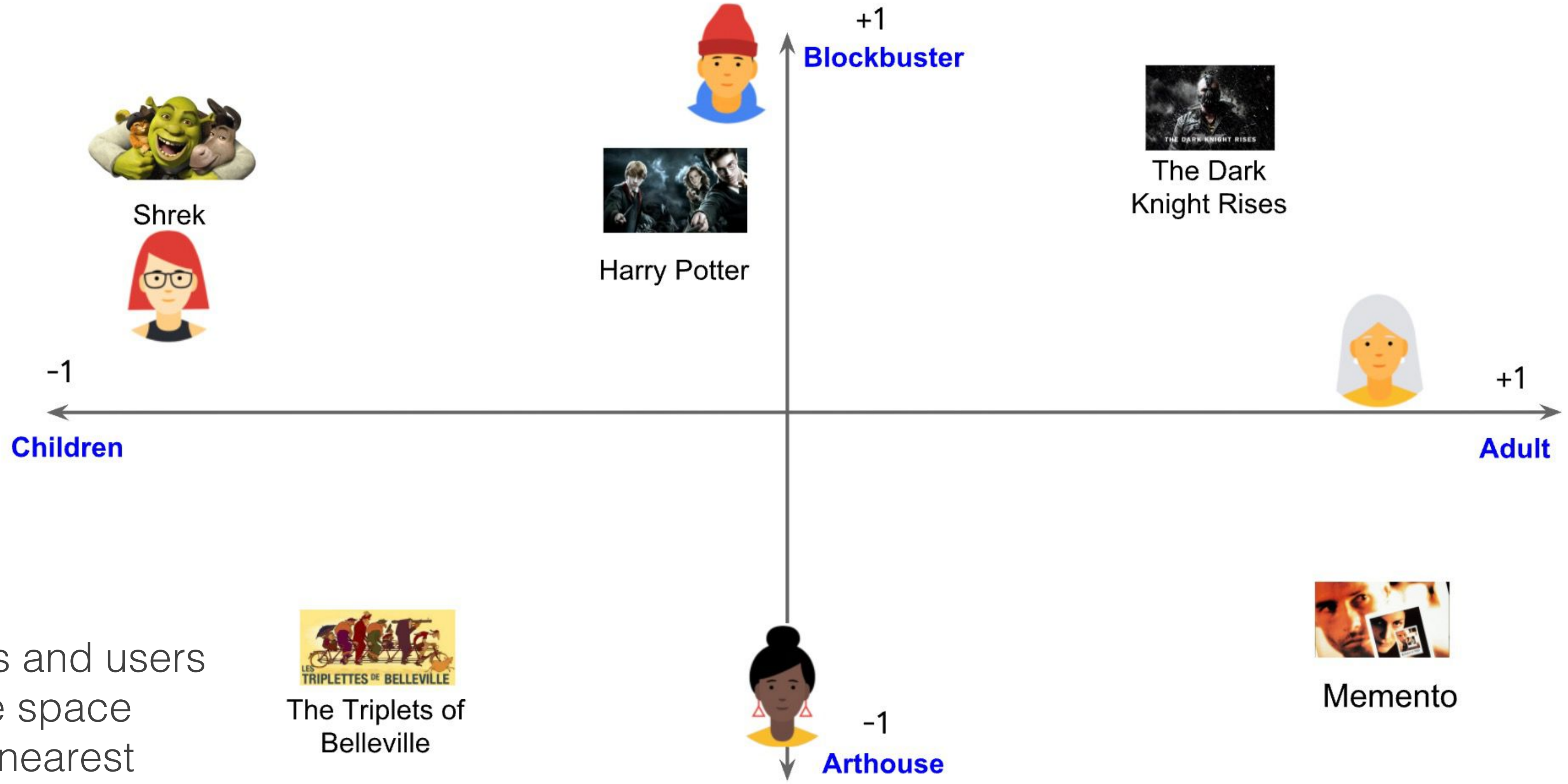
	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Issues:

1. How do we measure similarity? (e.g. correlation)
2. How many other users to consider?
3. How do we create the prediction?

Jannach, D., Zanker, M., Felfernig, A. and Friedrich, G., 2010. Recommender systems: an introduction. Cambridge University Press.

Recommender Systems: content-based filtering



Place movies and users into a feature space and find the nearest movies to the user's stated preferences

Image from https://developers.google.com/machine-learning/recommendation/collaborative/basics

Movies are assigned feature values based on their content (e.g. manually)

User provides ratings related to movie features

- These are used to make predictions of similar moview
- Other users' information is **not** used

Recommender Systems:
content-based filtering

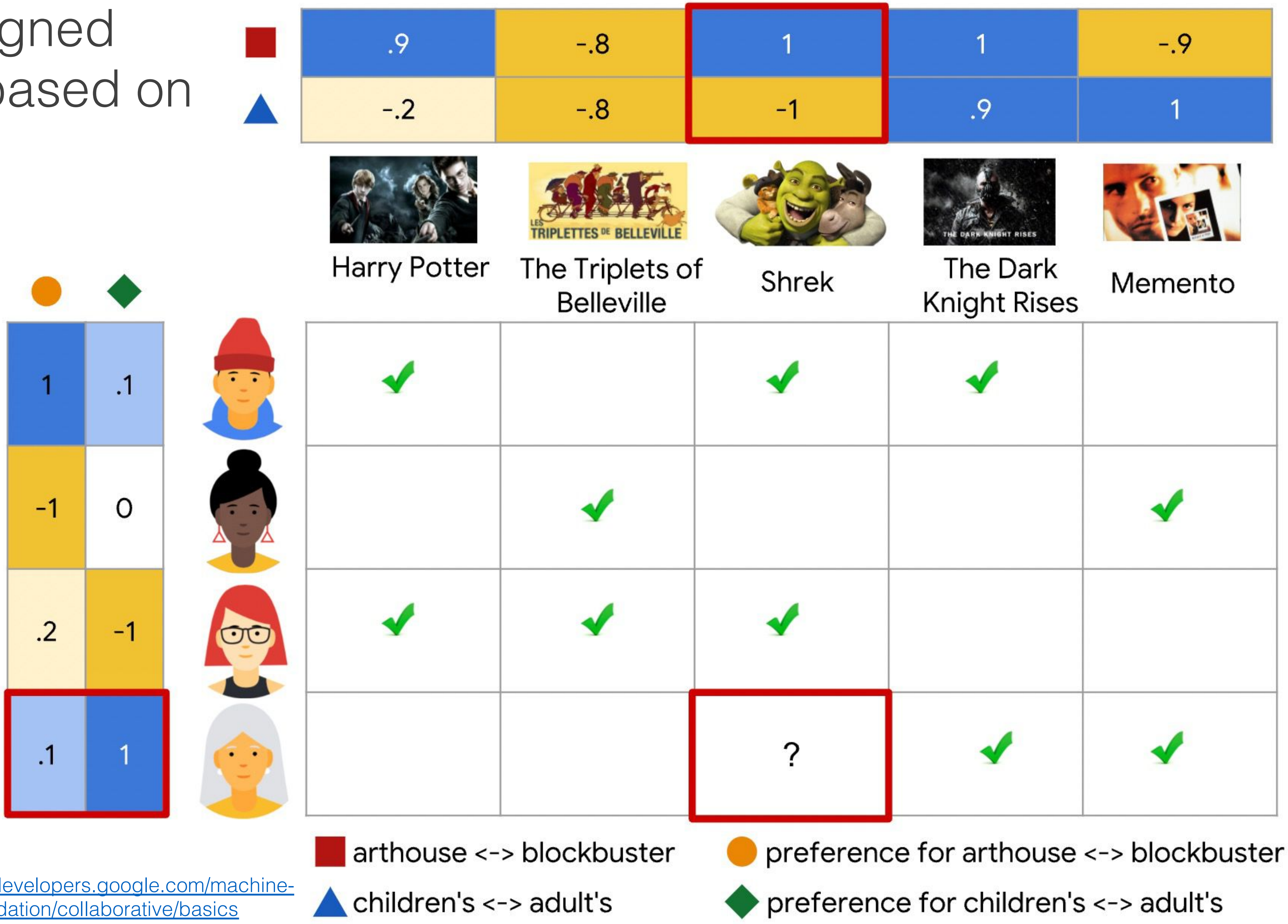
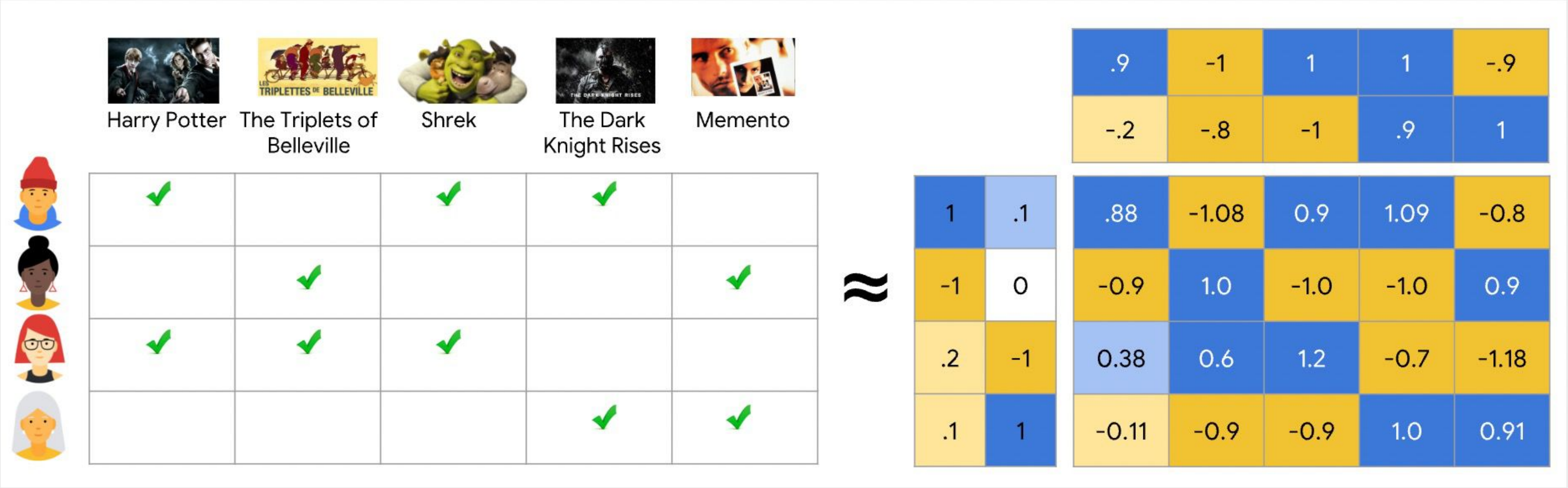


Image from https://developers.google.com/machine-learning/recommendation/collaborative/basics

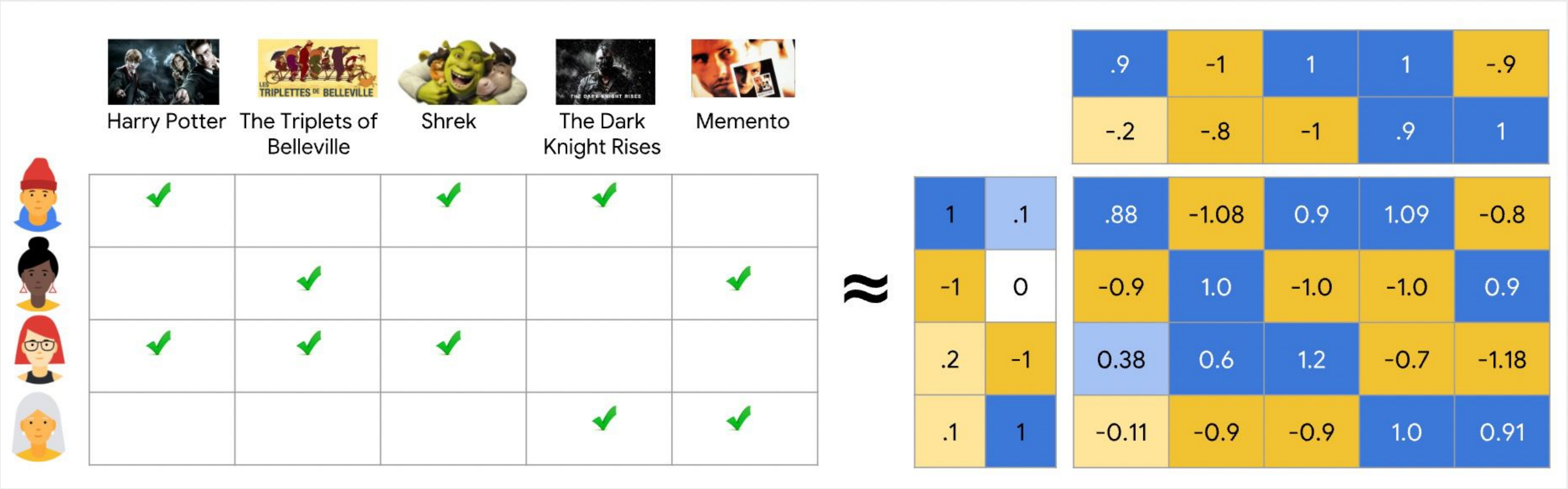
Recommender Systems: collaborative filtering



1. Find movies “similar” to a movie being evaluated based on user preferences and use those similar movies to estimate rating
2. Find users “similar” to the user based on other movie ratings, and use their ratings of the movie to predict the rating

Image from https://developers.google.com/machine-learning/recommendation/collaborative/basics

Recommender Systems: collaborative filtering



We can estimate the “features” of the movies (a.k.a. latent factors) through a decomposition of the sparse ratings matrix (e.g. through the singular value decomposition)

Image from https://developers.google.com/machine-learning/recommendation/collaborative/basics

Special Topics

Semi-supervised learning

Self-supervised learning

Recommender systems

Other Practical Considerations

Practical Considerations for Machine Learning

1. Let your problem/question drive your design choices
2. Set a reasonable goal and clear metric of success
3. Ask yourself if there are non-ML approaches that would work
4. Develop an end-to-end pipeline as soon as you're able and keep it maintained (data preparation & preprocessing, analysis, and performance evaluation)
5. Start with the simplest solution you can and layer on complexity as needed

- Features / representations are **often more important** than algorithms

Adapted from Google: https://developers.google.com/machine-learning/guides/rules-of-ml

More advice

- ALWAYS look at your data – before you begin, the inputs/outputs, etc.
- Check your distributions
- Explore outliers to get insights on the model
- Report confidence intervals whenever possible (make sure your “better” model is not just a noisy aberration)
- When comparing supervised models, make sure your comparing on the same validation set
- Make sure you NEVER mix training and validation information

Adapted from https://www.unofficialgoogledatascience.com/2016/10/practical-advice-for-analysis-of-large.html

References for further exploration

- Semi-supervised learning
- Self-supervised learning CVPR tutorial
- Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D. and Makedon, F., 2020. A survey on contrastive self-supervised learning. Technologies, 9(1), p.2. (link)
- Jannach, D., Zanker, M., Felfernig, A. and Friedrich, G., 2010. Recommender systems: an introduction. Cambridge University Press.