Deep Classifier Mimicry without Data Access



Steven Braun[1]   Martin Mundt[1,2]   Kristian Kersting[1,2,3,4]

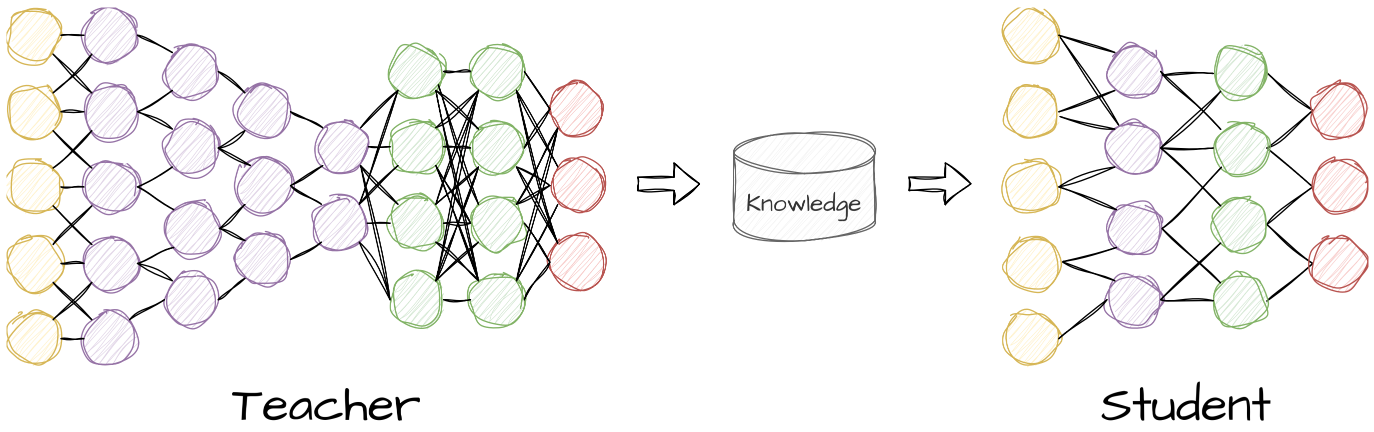
TECHNISCHE UNIVERSITÄT DARMSTADT

[1]Department of Computer Science, TU Darmstadt
[2]Hessian Center for AI (hessian.AI)
[3]German Research Center for Artificial Intelligence (DFKI)
[4]Centre for Cognitive Science, TU Darmstadt

hessian.AI

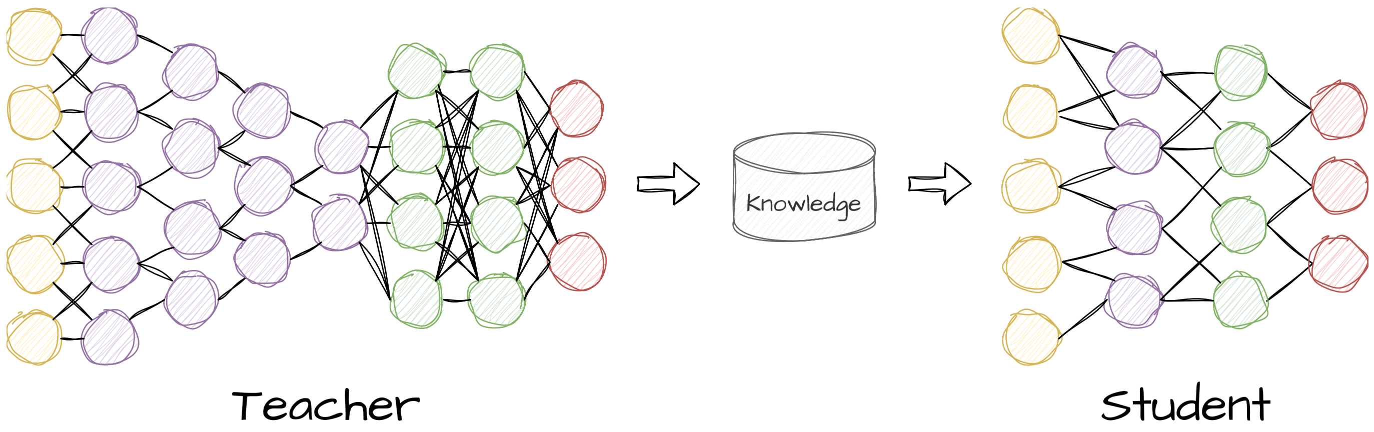
KompAKI

What is Knowledge Distillation?



[1]Hinton, G.E., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. ArXiv, abs/1503.02531.

What is Knowledge Distillation?

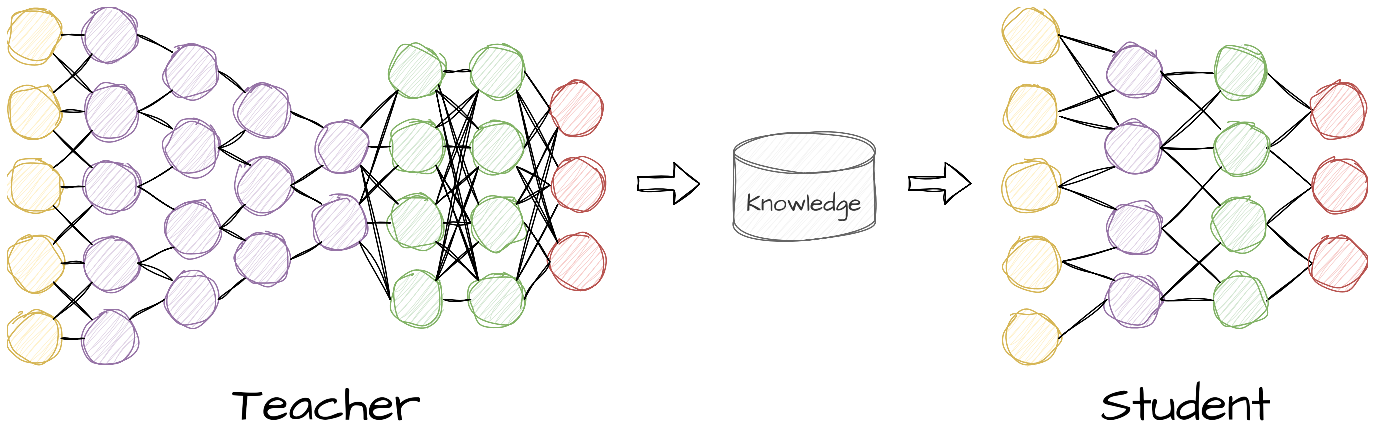


Original Formulation by Hinton et al.[1]: Teacher f^T, student f^S

$$\mathcal{L}_{\text{KD}}(\boldsymbol{x}, y) = \lambda \underbrace{\mathcal{L}_{\text{hard}}\left(f^S(\boldsymbol{x}), y\right)}_{\text{match data}} + (1-\lambda) \underbrace{\mathcal{L}_{\text{soft}}\left(f^S(\boldsymbol{x}), f^T(\boldsymbol{x})\right)}_{\text{match teacher}}$$

[1]Hinton, G.E., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. ArXiv, abs/1503.02531.

What is Knowledge Distillation?



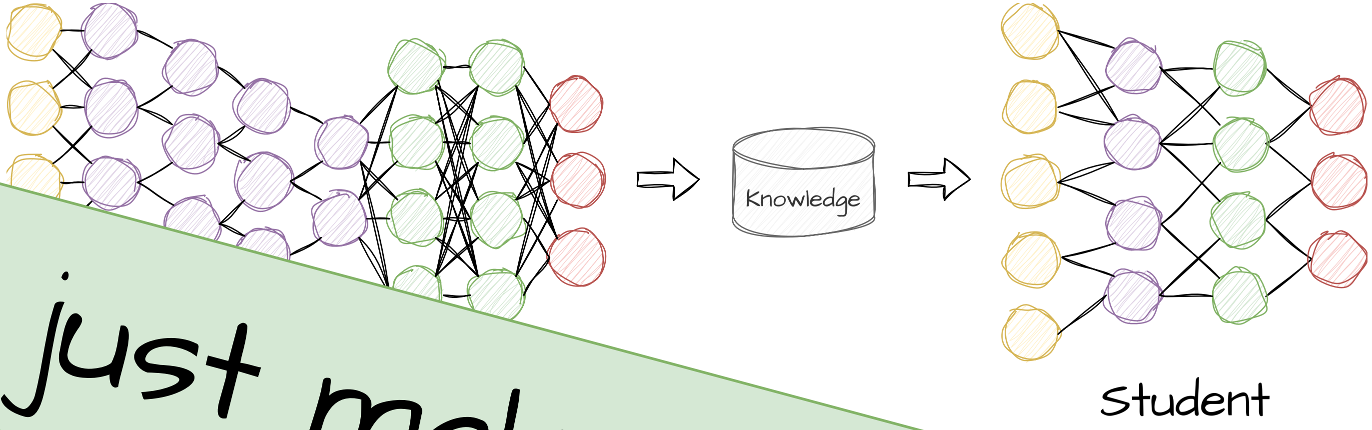
Original Formulation by Hinton et al.[1]: Teacher f^T, student f^S

$$\mathcal{L}_{\mathrm{KD}}(\boldsymbol{x}, y) = \lambda \underbrace{\mathcal{L}_{\mathrm{hard}}\left(f^S(\boldsymbol{x}), y\right)}_{\text{match data}} + (1-\lambda) \underbrace{\mathcal{L}_{\mathrm{soft}}\left(f^S(\boldsymbol{x}), f^T(\boldsymbol{x})\right)}_{\text{match teacher}}$$

↪ What if we don't have access to the **original training data** $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}$?

[1]Hinton, G.E., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. ArXiv, abs/1503.02531.

What is Knowledge Distillation?



We just make our own data!

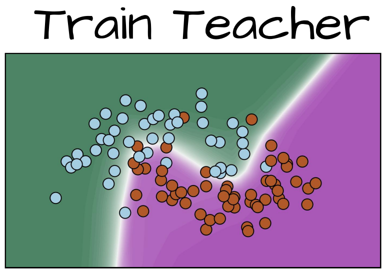
Original Formulation by Hin…

$$\mathcal{L}_{\text{KD}}(\boldsymbol{x}, y) = \lambda \underbrace{\mathcal{L}_{\text{hard}}\left(f^S(\boldsymbol{x}), y\right)}_{\text{match data}} \ldots$$

↪ What if we don't have access to the **original training data** $\mathcal{D} = \{(x_i, \ldots$

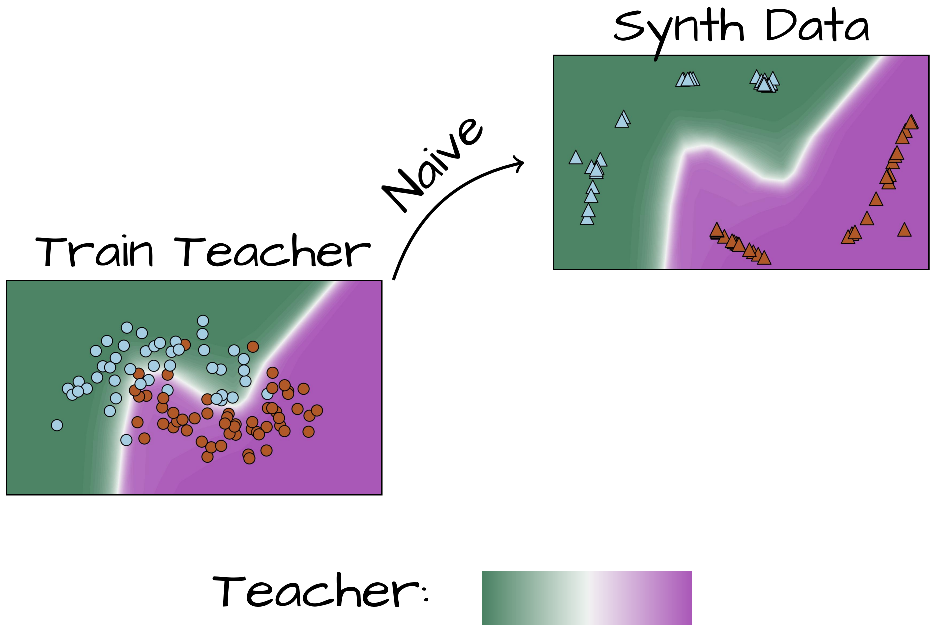
[1]Hinton, G.E., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. ArXiv, abs/1503.02531.

Extracting Knowledge



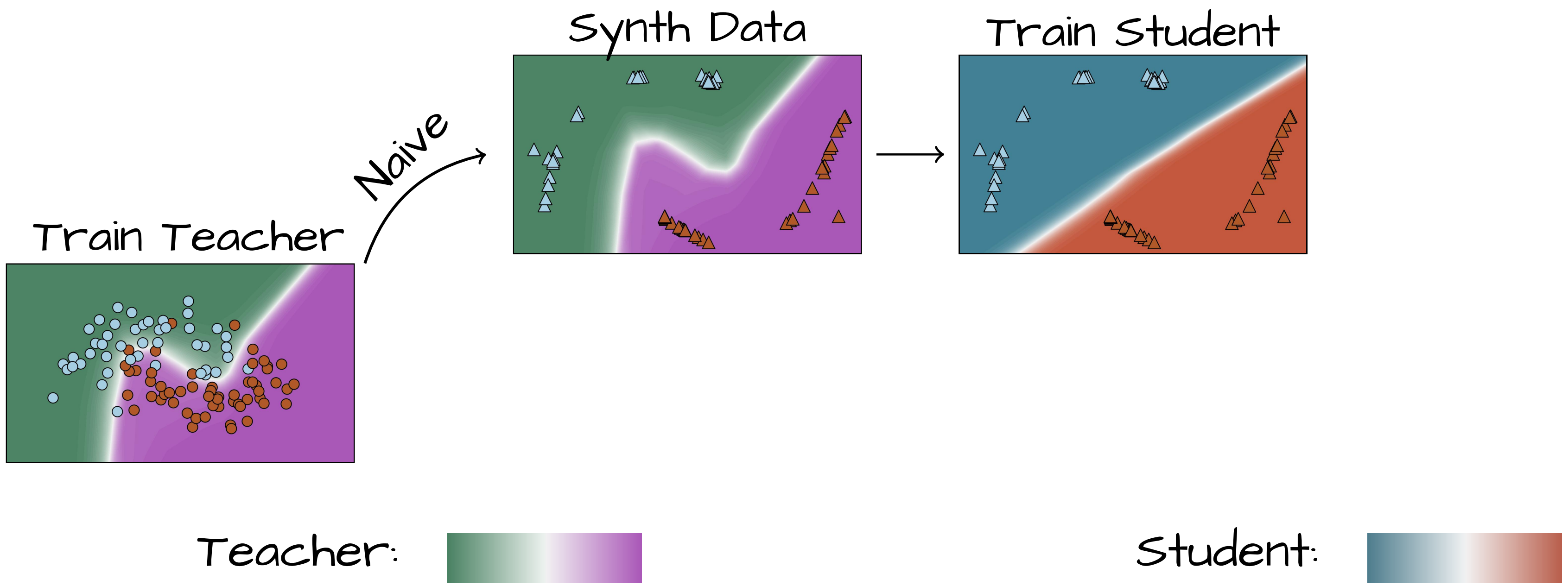
Extracting Knowledge

- **Naive**: Init. random datapoints $(\tilde{\boldsymbol{x}}, \tilde{y})$ and minimize $\mathcal{L}(\tilde{\boldsymbol{x}}, \tilde{y}) = \mathrm{CE}(f^T(\tilde{\boldsymbol{x}}), \tilde{y})$



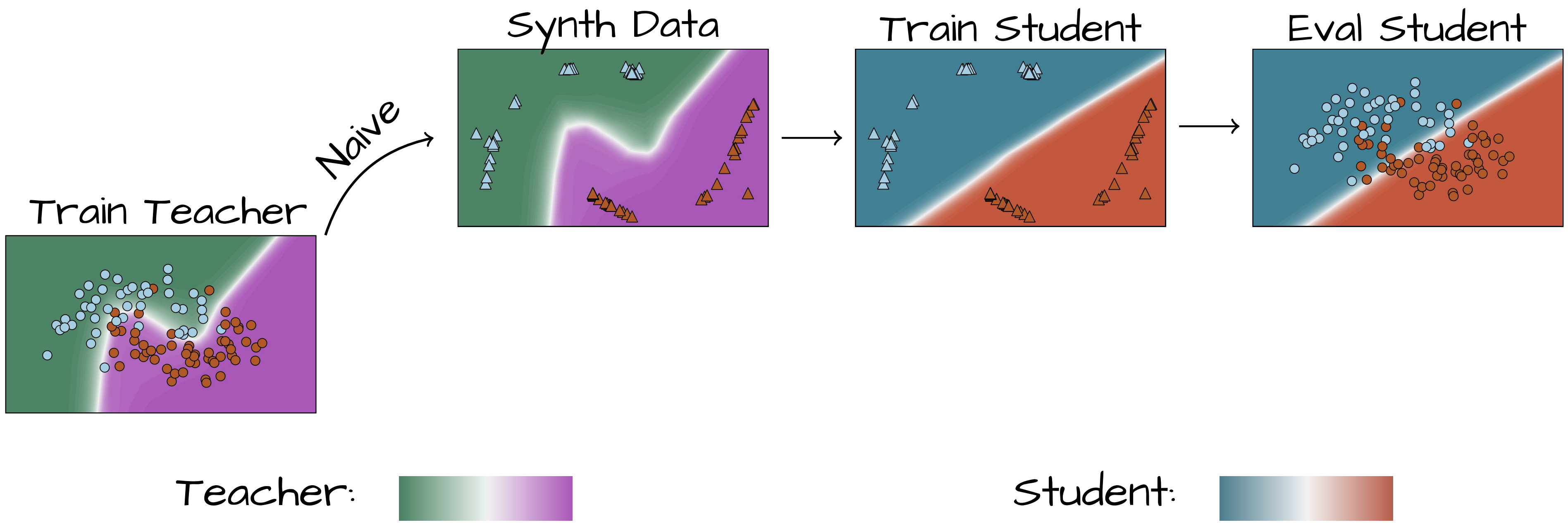
Extracting Knowledge

- **Naive:** Init. random datapoints $(\tilde{\boldsymbol{x}}, \tilde{y})$ and minimize $\mathcal{L}(\tilde{\boldsymbol{x}}, \tilde{y}) = \mathrm{CE}(f^T(\tilde{\boldsymbol{x}}), \tilde{y})$



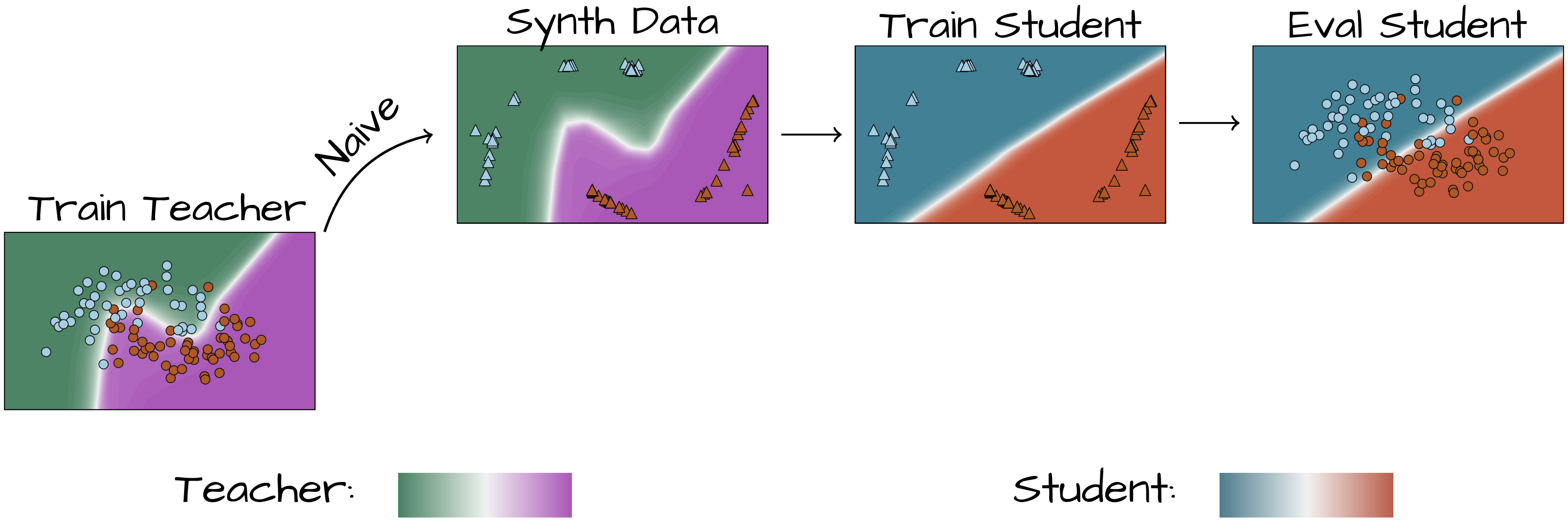
Extracting Knowledge

- **Naive:** Init. random datapoints $(\tilde{\boldsymbol{x}}, \tilde{y})$ and minimize $\mathcal{L}(\tilde{\boldsymbol{x}}, \tilde{y}) = \mathrm{CE}(f^T(\tilde{\boldsymbol{x}}), \tilde{y})$



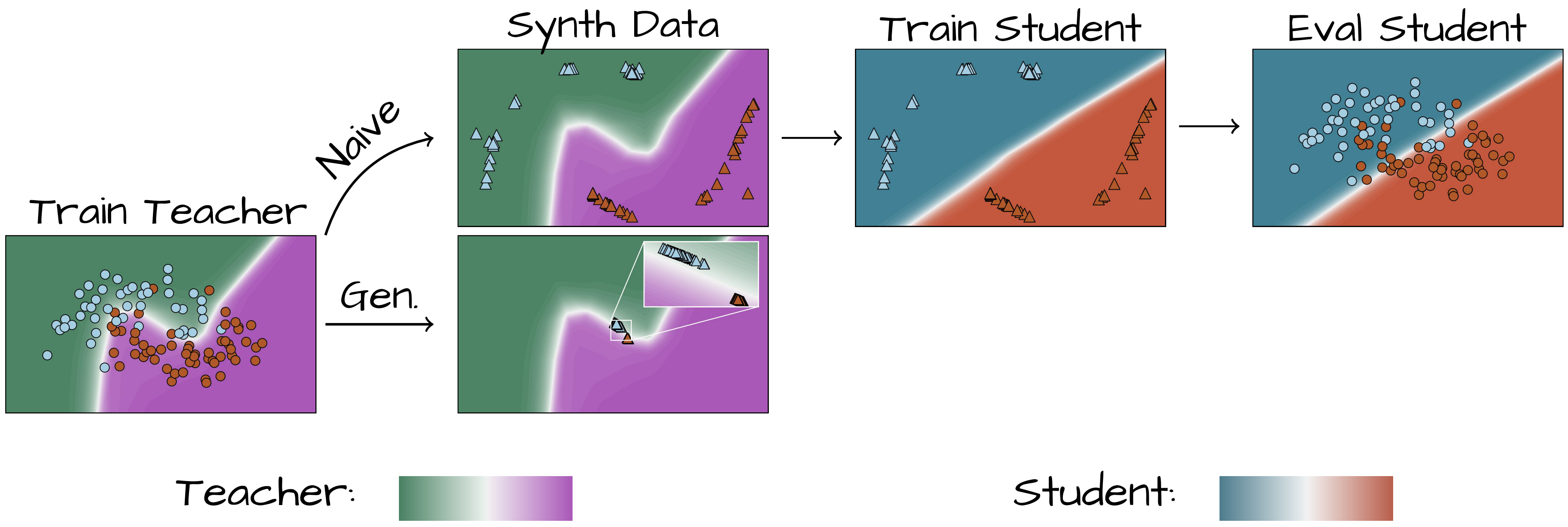
Extracting Knowledge

- **Generative:** Init. random latents $(\tilde{z}, \tilde{y})$ and minimize $\mathcal{L}(g_\theta(\tilde{z}), \tilde{y}) = \mathrm{CE}(f^T(g_\theta(\tilde{z})), \tilde{y})$



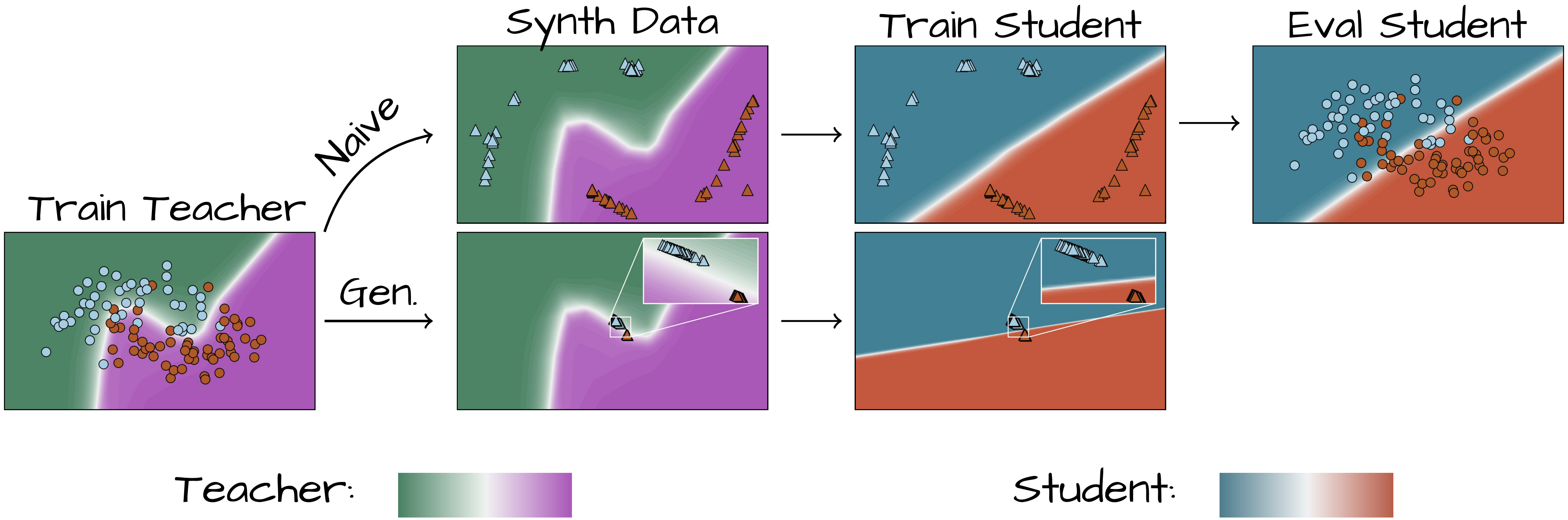
Extracting Knowledge

- **Generative:** Init. random latents $(\tilde{z}, \tilde{y})$ and minimize $\mathcal{L}(g_\theta(\tilde{z}), \tilde{y}) = \mathrm{CE}(f^T(g_\theta(\tilde{z})), \tilde{y})$



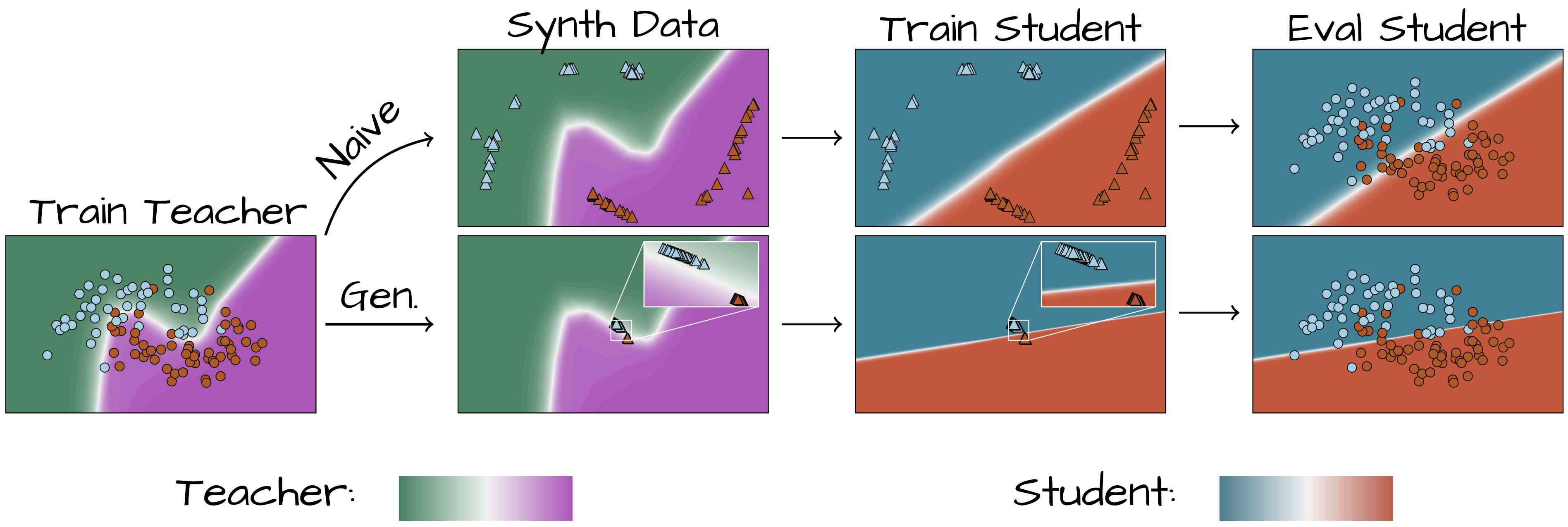
Extracting Knowledge

- **Generative:** Init. random latents $(\tilde{z}, \tilde{y})$ and minimize $\mathcal{L}(g_\theta(\tilde{z}), \tilde{y}) = \mathrm{CE}(f^T(g_\theta(\tilde{z})), \tilde{y})$



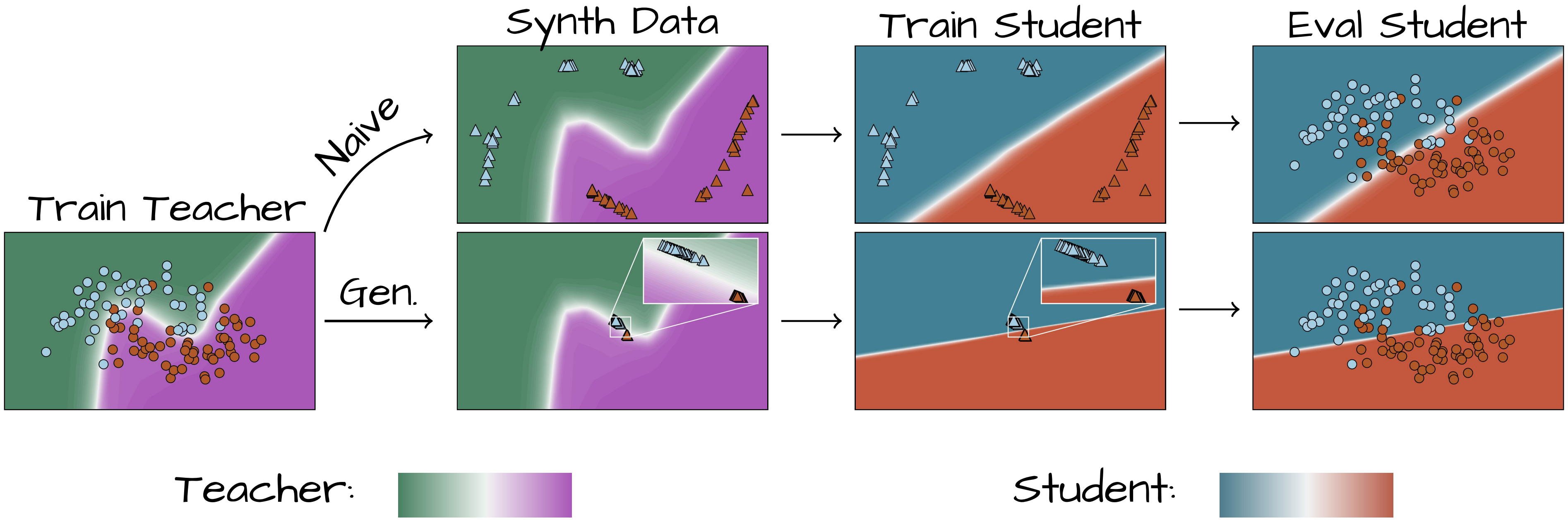
Extracting Knowledge

- **Generative:** Init. random latents $(\tilde{z}, \tilde{y})$ and minimize $\mathcal{L}(g_\theta(\tilde{z}), \tilde{y}) = \mathrm{CE}(f^T(g_\theta(\tilde{z})), \tilde{y})$



Extracting Knowledge

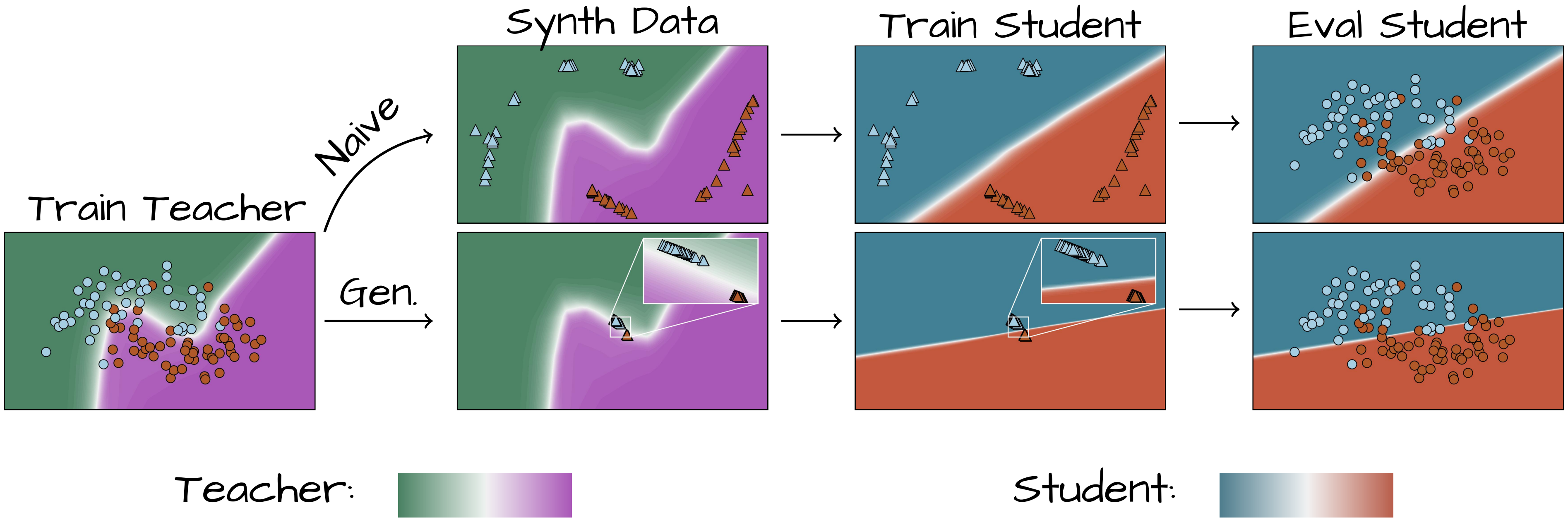
- **Generative:** Init. random latents $(\tilde{z}, \tilde{y})$ and minimize $\mathcal{L}(g_\theta(\tilde{z}), \tilde{y}) = \mathrm{CE}(f^T(g_\theta(\tilde{z})), \tilde{y})$



What's missing?

Extracting Knowledge

- **Generative:** Init. random latents $(\tilde{z}, \tilde{y})$ and minimize $\mathcal{L}(g_\theta(\tilde{z}), \tilde{y}) = \mathrm{CE}(f^T(g_\theta(\tilde{z})), \tilde{y})$



What's missing?

Naive: Keep classes close, else boundary becomes linear

Generative: Disperse samples along the relevant boundary region

CAKE: Contrastive Abductive Knowledge Extraction

Idea: Contrast sample pairs noisily *across* and *along* the *relevant* teacher decision boundary and regularize with data priors!

CAKE: Contrastive Abductive Knowledge Extraction

Idea: Contrast sample pairs noisily *across* and *along* the *relevant* teacher decision boundary and regularize with data priors!

- Contrastive samples between classes

$$\mathcal{L}_{\text{contr}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathbb{1}\left[y_i \neq y_j\right] \, \|\boldsymbol{f}^T(\boldsymbol{x}_i) - \boldsymbol{f}^T(\boldsymbol{x}_j)\|_2^2$$

... or any other contrastive loss

CAKE: Contrastive Abductive Knowledge Extraction

Idea: Contrast sample pairs noisily *across* and *along* the *relevant* teacher decision boundary and regularize with data priors!

- Contrastive samples between classes

$$\mathcal{L}_{\text{contr}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathbb{1}\left[y_i \neq y_j\right] \, \|\boldsymbol{f}^T(\boldsymbol{x}_i) - \boldsymbol{f}^T(\boldsymbol{x}_j)\|_2^2$$

... or any other contrastive loss

- Regularize using domain knowledge

$$\mathcal{L}_{\text{TV}}(\boldsymbol{x}) = \sum_{j,k} \|\boldsymbol{x}_{j,k} - \boldsymbol{x}_{j-1,k}\| + \|\boldsymbol{x}_{j,k} - \boldsymbol{x}_{j,k-1}\|$$

... or any other data prior

CAKE: Contrastive Abductive Knowledge Extraction

Idea: Contrast sample pairs noisily *across* and *along* the *relevant* teacher decision boundary and regularize with data priors!

- Contrastive samples between classes

$$\mathcal{L}_{\text{contr}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathbb{1}\left[y_i \neq y_j\right] \, \|\boldsymbol{f}^T(\boldsymbol{x}_i) - \boldsymbol{f}^T(\boldsymbol{x}_j)\|_2^2$$

... or any other contrastive loss

- Regularize using domain knowledge

$$\mathcal{L}_{\text{TV}}(\boldsymbol{x}) = \sum_{j,k} \|\boldsymbol{x}_{j,k} - \boldsymbol{x}_{j-1,k}\| + \|\boldsymbol{x}_{j,k} - \boldsymbol{x}_{j,k-1}\|$$

... or any other data prior

- Noisily disperse samples along the boundary

CAKE: Contrastive Abductive Knowledge Extraction

Idea: Contrast sample pairs noisily *across* and *along* the *relevant* teacher decision boundary and regularize with data priors!

- Contrastive samples between classes

$$\mathcal{L}_{\text{contr}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathbb{1}[y_i \neq y_j] \; \|\boldsymbol{f}^T(\boldsymbol{x}_i) - \boldsymbol{f}^T(\boldsymbol{x}_j)\|_2^2$$

... or any other contrastive loss

- Regularize using domain knowledge

$$\mathcal{L}_{\text{TV}}(\boldsymbol{x}) = \sum_{j,k} \|\boldsymbol{x}_{j,k} - \boldsymbol{x}_{j-1,k}\| + \|\boldsymbol{x}_{j,k} - \boldsymbol{x}_{j,k-1}\|$$

... or any other data prior

- Noisily disperse samples along the boundary

Explicit: Langevin Dynamics $\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \nabla_x \mathcal{L}(\boldsymbol{x}_i^t)\eta(t) + \sqrt{2\eta(t)}\varepsilon_i^t \text{ , with } \varepsilon_i^t \sim N(0, I)$

CAKE: Contrastive Abductive Knowledge Extraction

Idea: Contrast sample pairs noisily *across* and *along* the *relevant* teacher decision boundary and regularize with data priors!

- Contrastive samples between classes

$$\mathcal{L}_{\text{contr}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathbb{1}\left[y_i \neq y_j\right] \, \|\boldsymbol{f}^T(\boldsymbol{x}_i) - \boldsymbol{f}^T(\boldsymbol{x}_j)\|_2^2$$

... or any other contrastive loss

- Regularize using domain knowledge

$$\mathcal{L}_{\text{TV}}(\boldsymbol{x}) = \sum_{j,k} \|\boldsymbol{x}_{j,k} - \boldsymbol{x}_{j-1,k}\| + \|\boldsymbol{x}_{j,k} - \boldsymbol{x}_{j,k-1}\|$$

... or any other data prior

- Noisily disperse samples along the boundary

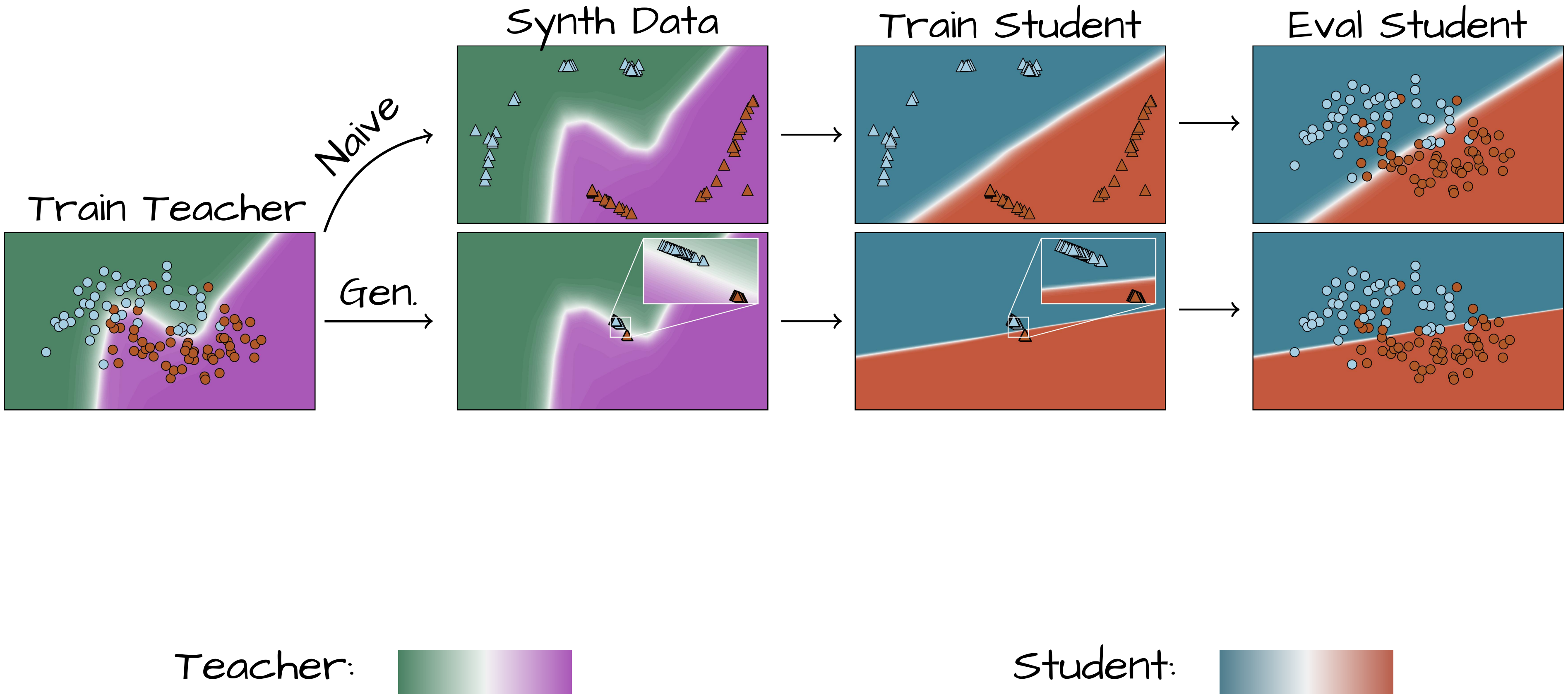
Explicit: Langevin Dynamics $\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \nabla_x \mathcal{L}(\boldsymbol{x}_i^t)\eta(t) + \sqrt{2\eta(t)}\varepsilon_i^t$, with $\varepsilon_i^t \sim N(0, I)$

Implicit: Stochasticity of SGD and step size schedules $\eta(t)$ is enough

... or any other noise injection

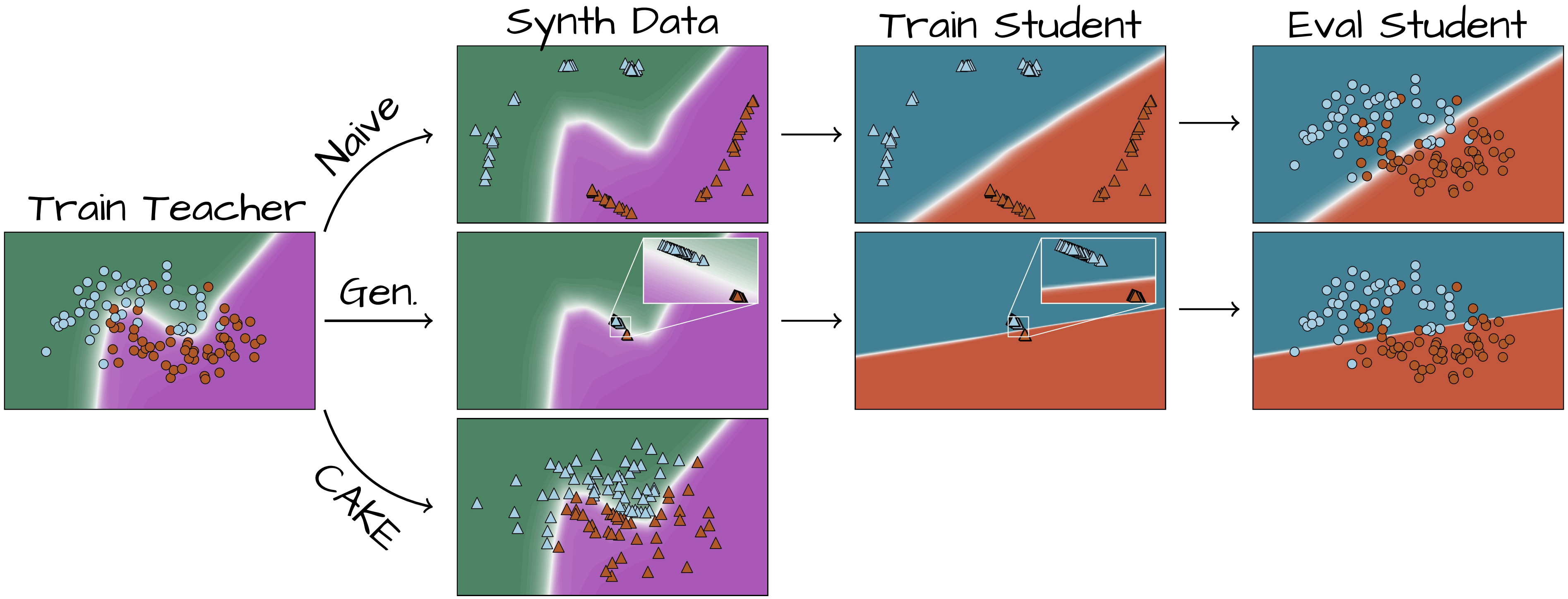
Extracting Knowledge

- **CAKE**: Contrast pairs noisily across and along the relevant teacher decision boundary.



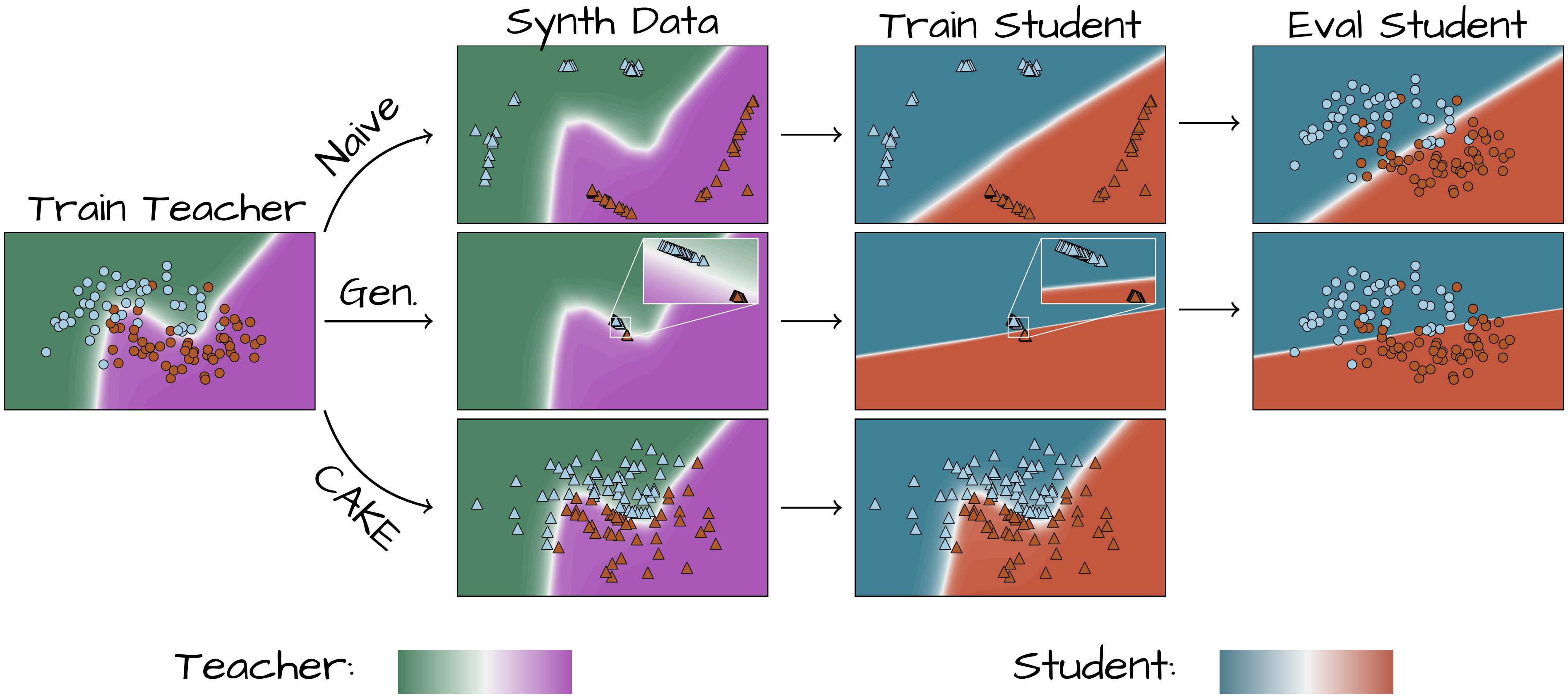
Extracting Knowledge

- **CAKE**: Contrast pairs noisily across and along the relevant teacher decision boundary.



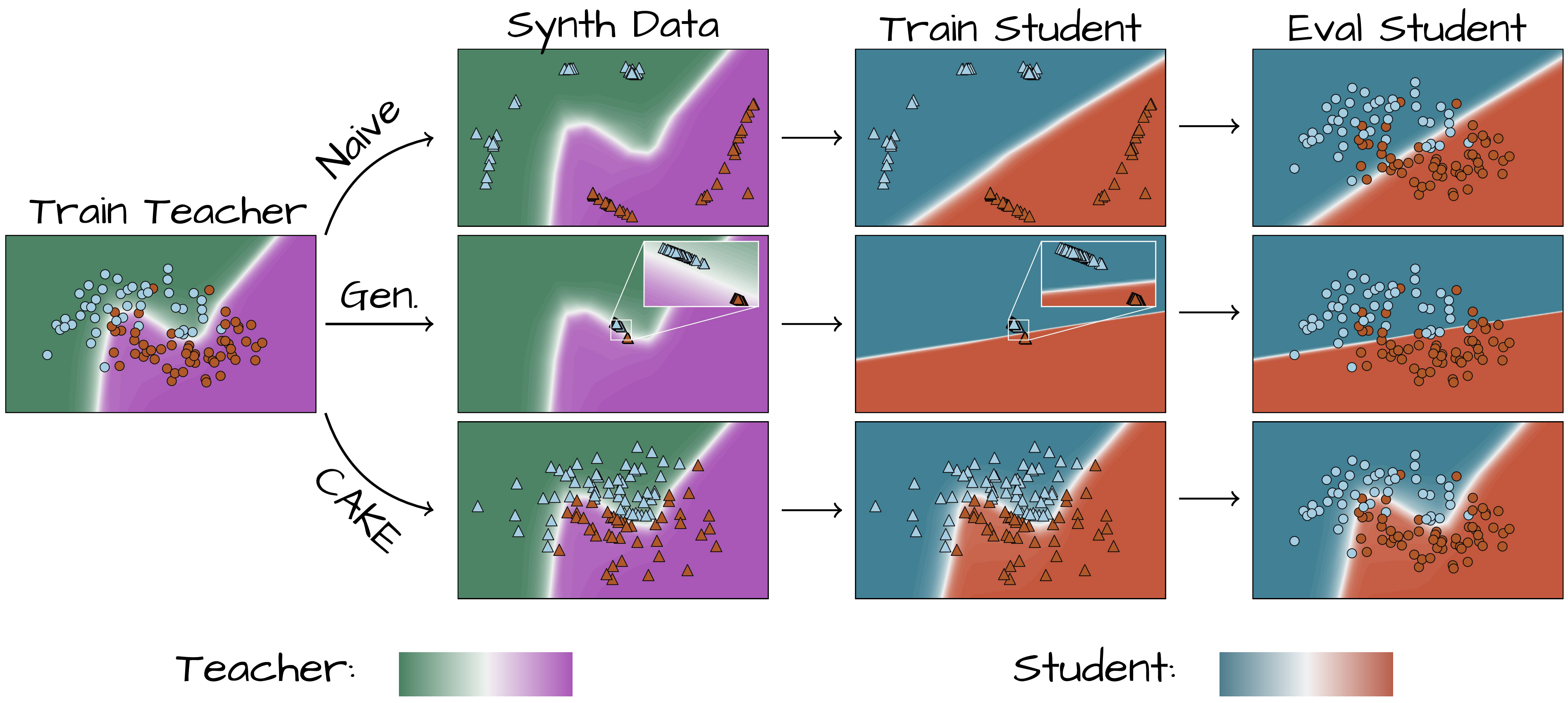
Extracting Knowledge

- **CAKE**: Contrast pairs noisily across and along the relevant teacher decision boundary.



Extracting Knowledge

- **CAKE**: Contrast pairs noisily across and along the relevant teacher decision boundary.



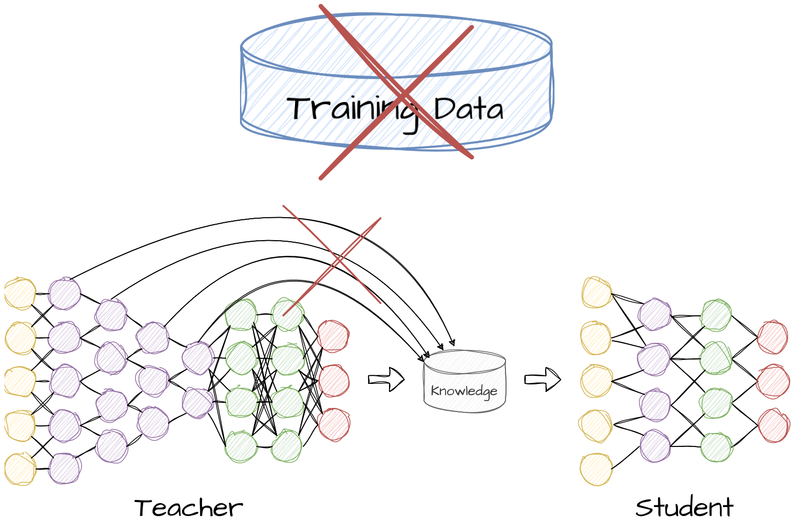
CAKE Lifts Knowledge Distillation Restrictions

- **No original data access**

CAKE Lifts Knowledge Distillation Restrictions

- **No original data access**

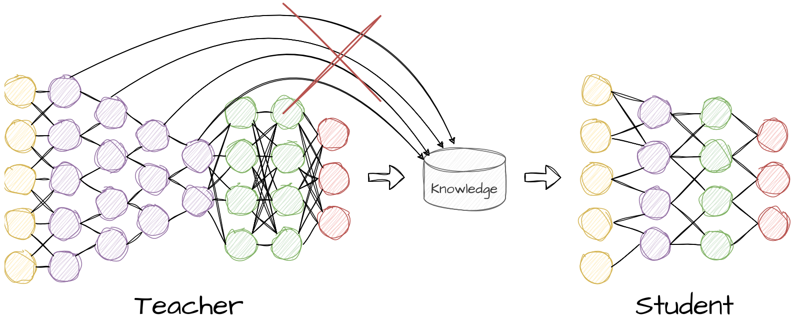
- **No model access**
  e.g. intermediate activations



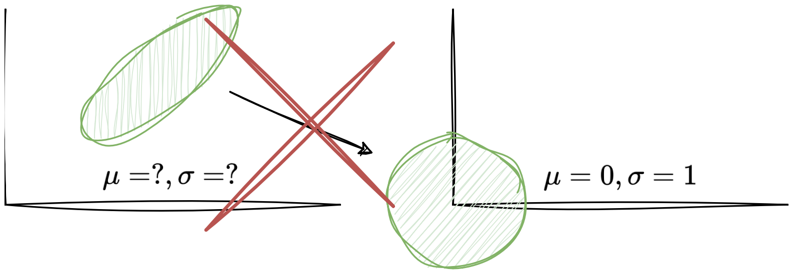
CAKE Lifts Knowledge Distillation Restrictions

- **No original data access**

- **No model access**
  e.g. intermediate activations



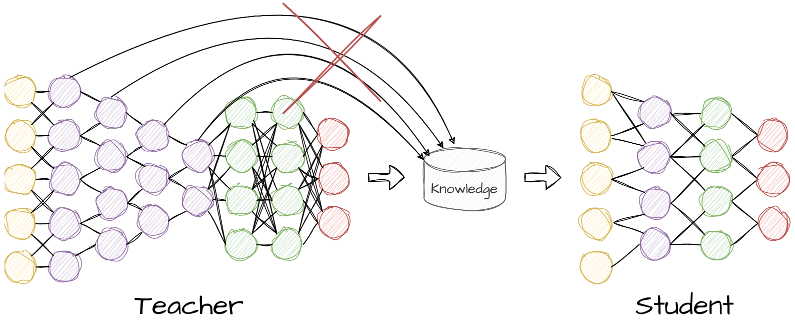
- **No model assumptions**
  e.g. BatchNorm, linear penultimate layer



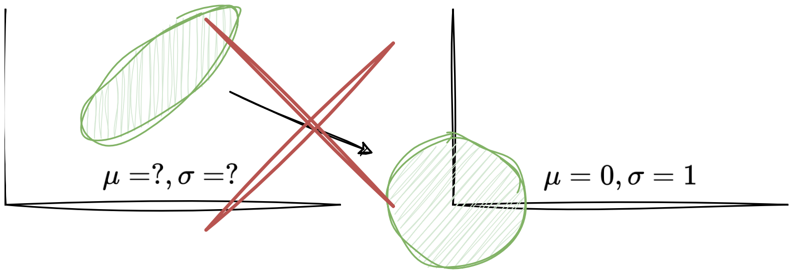
CAKE Lifts Knowledge Distillation Restrictions

- **No original data access**

- **No model access**
  e.g. intermediate activations



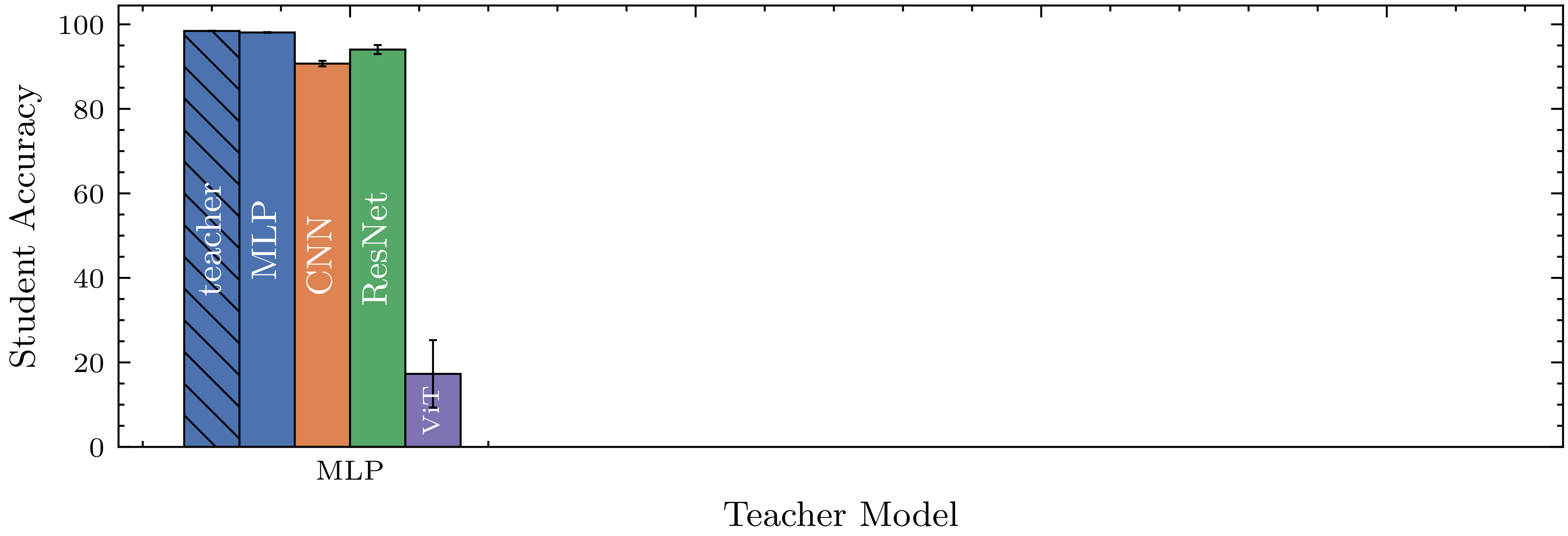
- **No model assumptions**
  e.g. BatchNorm, linear penultimate layer



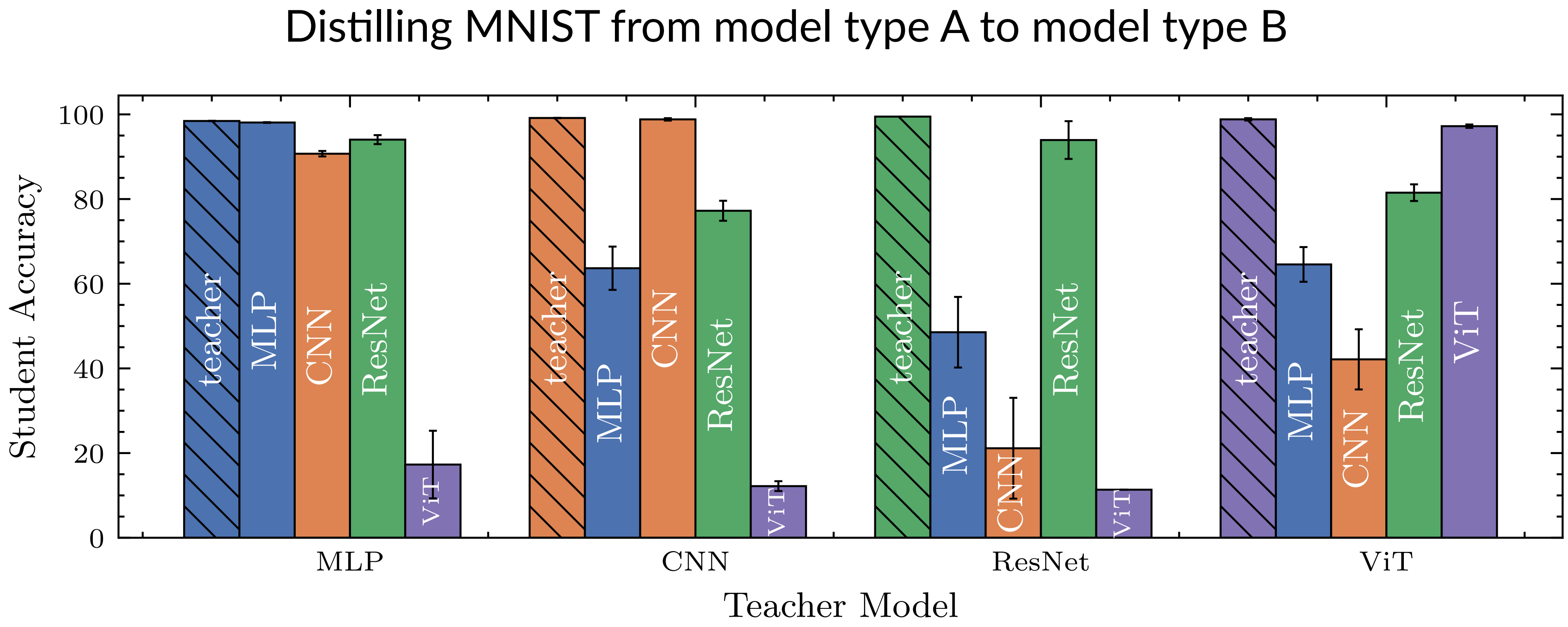
↪ CAKE can be applied to any “blackbox” model which is differentiable w.r.t. its input.

CAKE Across Model Types

Distilling MNIST from model type A to model type B



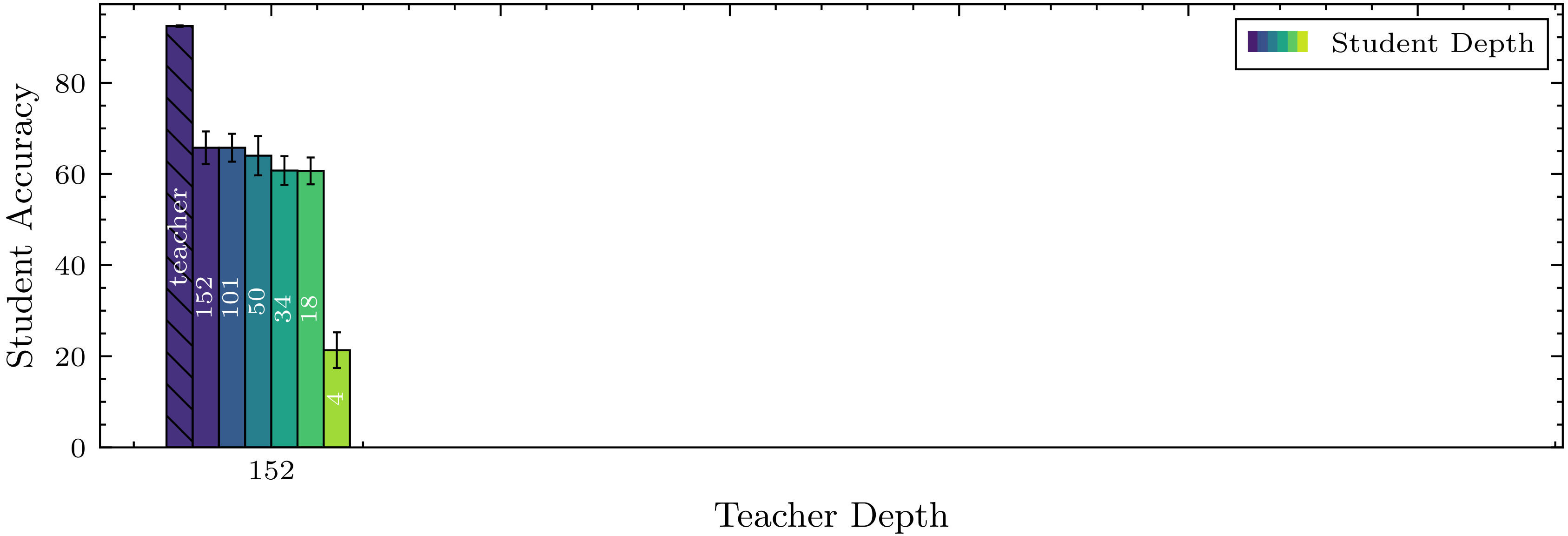
CAKE Across Model Types



Takeaways: 1. Similar inductive bias → better distillation
2. Less inductive bias → better distillation 3. ResNet is a safe student model choice.

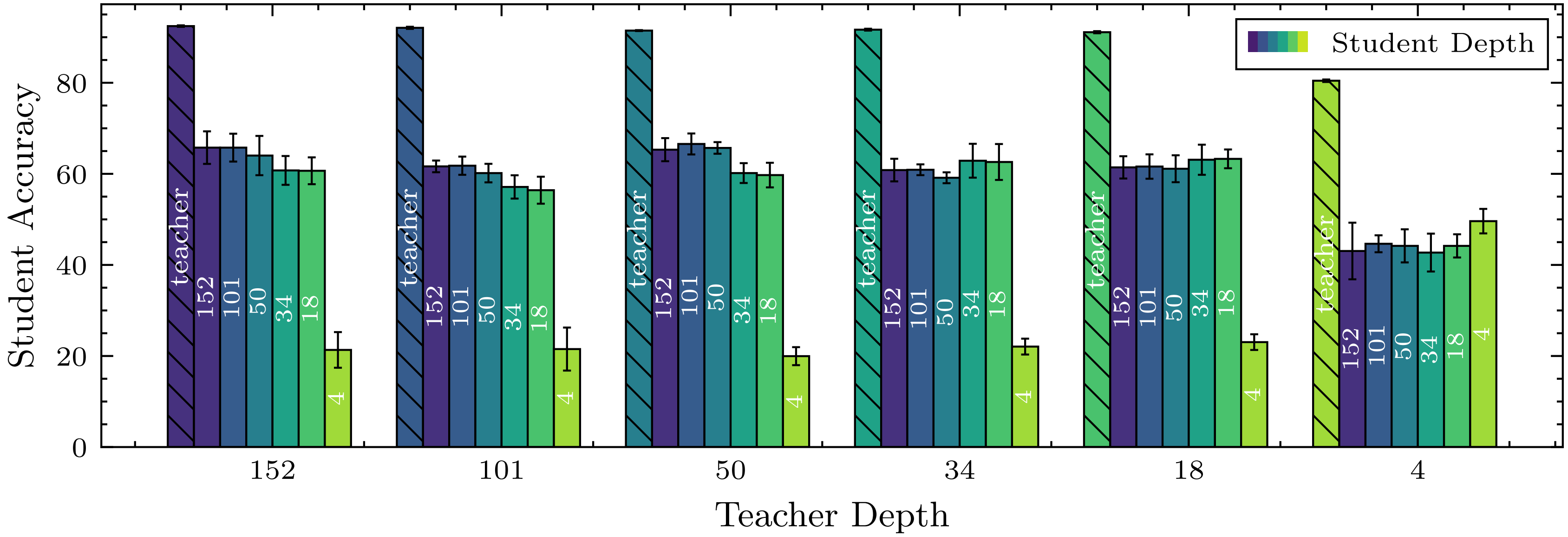
CAKE Across Scales

Distilling CIFAR-10 knowledge from ResNet-X to ResNet-Y (152, 101, 50, 34, 18, 4)



CAKE Across Scales

Distilling CIFAR-10 knowledge from ResNet-X to ResNet-Y (152, 101, 50, 34, 18, 4)

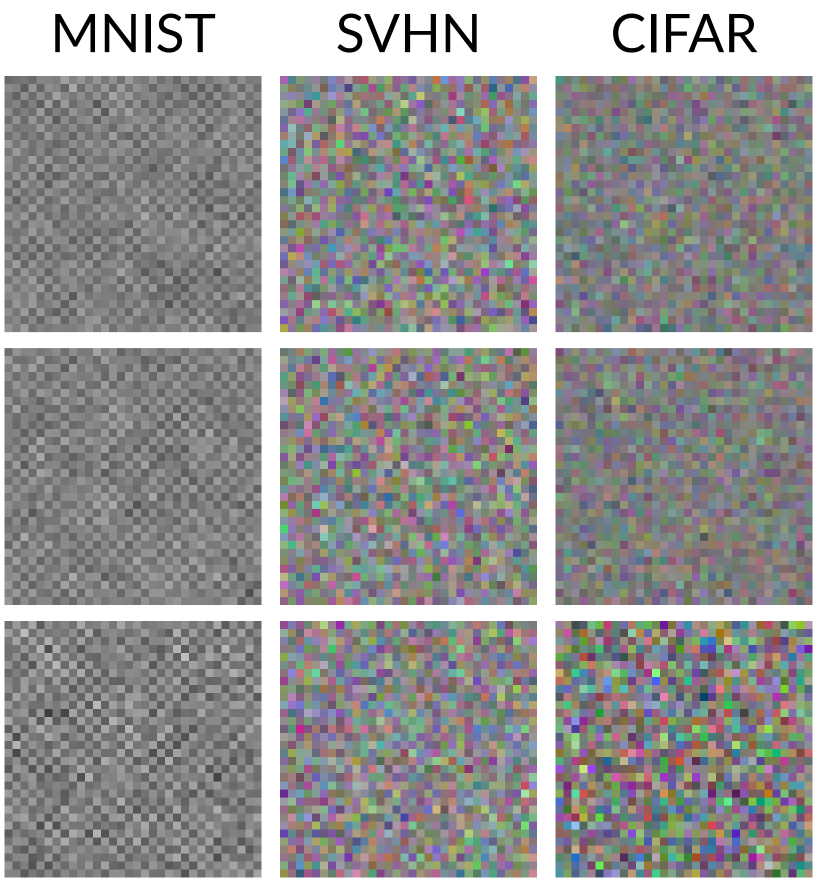


Takeaway: CAKE can compress models at a stable accuracy until capacity is too heavily constrained.

CAKE Synthetic Samples

No visual resemblance with original training data.
Possible future work includes:

- Differential privacy?
- Data utility and privacy trade-offs?
- Robustness against adversarial attacks?



Summary and Outlook

CAKE is a **data-free** and **model-agnostic** knowledge distillation method, that …

- can distill models *across scales*
- can distill between *different model types*
- doesn't produce data-like samples (visually)

Summary and Outlook

CAKE is a **data-free** and **model-agnostic** knowledge distillation method, that ...

- can distill models *across scales*
- can distill between *different model types*
- doesn't produce data-like samples (visually)

Future work

- Estimate gradients? $\rightarrow$ truly "blackbox", API-model possible
- Investigate the data privacy perspective?
- Investigate explicit instead of implicit noise

Still interested?

Join me at Room 2, Poster #117

Paper

Code

steven.braun@cs.tu-darmstadt.de