ICLR

# Does the Data Processing Inequality Reflect Practice? On the Utility of Low-Level Tasks

Roy Turgeman
Tom Tirer

Bar Ilan University

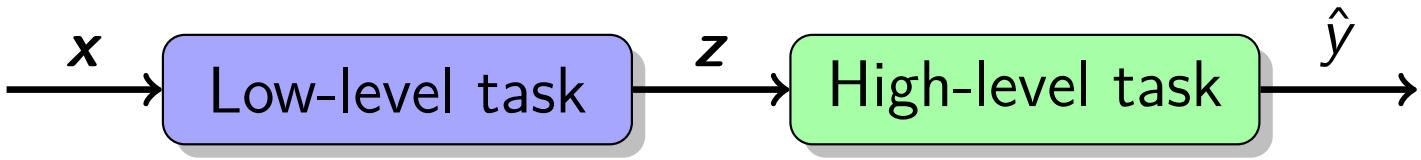
ICLR 2026

Contact

Paper

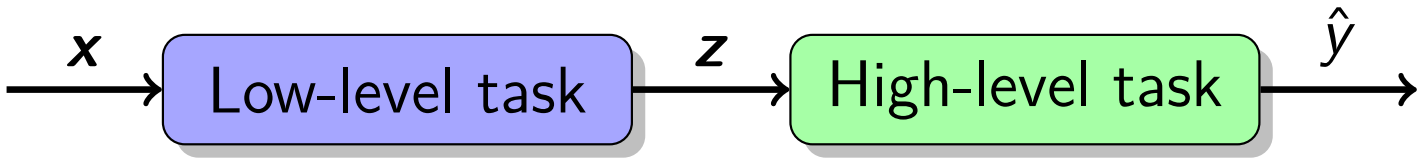
# Problem Addressed

- It is common practice to perform a low-level task (e.g., denoising, encoding, super-resolution) before addressing a high-level task (e.g., classification, detection).



# Problem Addressed

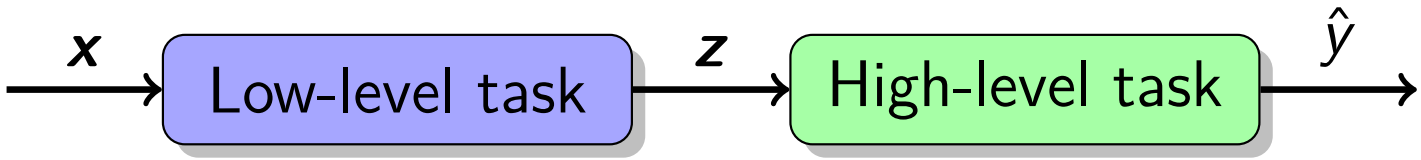
- It is common practice to perform a low-level task (e.g., denoising, encoding, super-resolution) before addressing a high-level task (e.g., classification, detection).




- However, this raises a theoretical question about when and why this actually helps the high-level task.

# Problem Addressed

- It is common practice to perform a low-level task (e.g., denoising, encoding, super-resolution) before addressing a high-level task (e.g., classification, detection).




- However, this raises a theoretical question about when and why this actually helps the high-level task.
- Example: super-resolution (low-level) and detection (high-level).

Figure: HR, LR (1/8), Bicubic-upsamp + detect, SR + detect (Haris, Shakhnarovich, and Ukita, 2021)

# The Bayes optimal classifier

For a Markov chain $y \to x \to z$, the data processing inequality states $I(x, y) \geq I(z, y)$. This concept transfers to optimal classification.

## Theorem 1

*Let $y \to x \to z$ be a Markov chain where $y \in \{1, 2\}$ denotes the sample class. We have*

$$\mathbb{P}(c_{opt}(x) \neq y) \leq \mathbb{P}(\tilde{c}_{opt}(z) \neq y),$$

*where $c_{opt}$ and $\tilde{c}_{opt}$ denote optimal Bayes classifiers.*

- Hence, in the case of binary classification, data processing cannot improve Bayes-optimal accuracy.

# The Bayes optimal classifier

For a Markov chain $y \to x \to z$, the data processing inequality states $I(x, y) \geq I(z, y)$. This concept transfers to optimal classification.

## Theorem 1

*Let $y \to x \to z$ be a Markov chain where $y \in \{1, 2\}$ denotes the sample class. We have*

$$\mathbb{P}(c_{opt}(x) \neq y) \leq \mathbb{P}(\tilde{c}_{opt}(z) \neq y),$$

*where $c_{opt}$ and $\tilde{c}_{opt}$ denote optimal Bayes classifiers.*

- Hence, in the case of binary classification, data processing cannot improve Bayes-optimal accuracy.
- A similar statement and proof can be found in an arXiv version of (Liu, Zhang, and Xiong, 2019).

# Our Main Contributions

- We theoretically show that in high-dimensional binary classification, pre-classification processing (e.g., dimensionality reduction) can improve accuracy, even for classifiers approaching the Bayes optimum.

# Our Main Contributions

- We theoretically show that in high-dimensional binary classification, pre-classification processing (e.g., dimensionality reduction) can improve accuracy, even for classifiers approaching the Bayes optimum.
- We analyze how factors like training size, SNR, and class imbalance affect the gain from processing. Surprisingly, the maximal gain can increase with greater SNR.

# Our Main Contributions

- We theoretically show that in high-dimensional binary classification, pre-classification processing (e.g., dimensionality reduction) can improve accuracy, even for classifiers approaching the Bayes optimum.
- We analyze how factors like training size, SNR, and class imbalance affect the gain from processing. Surprisingly, the maximal gain can increase with greater SNR.
- We extend the analysis to practical deep classifiers, studying the effects of denoising and self-supervised encoding on benchmark datasets under varying training sizes, noise levels, and class imbalance. The observed trends align with theoretical predictions.

# Data model

- Binary classification with a two-component GMM in $\mathbb{R}^d$.

$$y \in \{1, 2\}, \qquad \mathbf{x} \mid y = j \sim \mathcal{N}(\boldsymbol{\mu}_j, \sigma_j^2 \boldsymbol{I}_d), \qquad \mathbb{P}(y = j) = \pi_j.$$

# Data model

- Binary classification with a two-component GMM in $\mathbb{R}^d$.

$$y \in \{1, 2\}, \qquad \mathbf{x} \mid y = j \sim \mathcal{N}(\boldsymbol{\mu}_j, \sigma_j^2 \boldsymbol{I}_d), \qquad \mathbb{P}(y = j) = \pi_j.$$

- We further assume that

$$\boldsymbol{\mu}_2 = -\boldsymbol{\mu}_1 = \boldsymbol{\mu}, \qquad \sigma_1^2 = \sigma_2^2 = \sigma^2, \qquad \pi_1 = \pi_2 = 1/2,$$

where the magnitudes of the entries of $\boldsymbol{\mu}$ are bounded by some universal constant, and $\sigma$ is independent of $d$.

# Data model

- Binary classification with a two-component GMM in $\mathbb{R}^d$.

$$y \in \{1, 2\}, \qquad \mathbf{x} \mid y = j \sim \mathcal{N}(\boldsymbol{\mu}_j, \sigma_j^2 \boldsymbol{I}_d), \qquad \mathbb{P}(y = j) = \pi_j.$$

- We further assume that

$$\boldsymbol{\mu}_2 = -\boldsymbol{\mu}_1 = \boldsymbol{\mu}, \qquad \sigma_1^2 = \sigma_2^2 = \sigma^2, \qquad \pi_1 = \pi_2 = 1/2,$$

  where the magnitudes of the entries of $\boldsymbol{\mu}$ are bounded by some universal constant, and $\sigma$ is independent of $d$.
- The considered setup is standard in theoretical works that aim at rigorous mathematical analysis (Kothapalli and Tirer, 2025).

# Data model

- We define the GMM separation quality factor (interpreted as SNR):

$$\mathcal{S} := \left( \frac{\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|}{\sigma_1 + \sigma_2} \right)^2 = \frac{\|\boldsymbol{\mu}\|^2}{\sigma^2}.$$

  In contrast to many theoretical papers, our analysis covers SNR arbitrarily close to zero.

# Data model

- We define the GMM separation quality factor (interpreted as SNR):

$$\mathcal{S} := \left( \frac{\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|}{\sigma_1 + \sigma_2} \right)^2 = \frac{\|\boldsymbol{\mu}\|^2}{\sigma^2}.$$

  In contrast to many theoretical papers, our analysis covers SNR arbitrarily close to zero.

- The training data consists of $N_j$ labeled i.i.d. samples per class $j$, denoted by $\mathcal{D} = \{\mathbf{x}_{i,j} : j \in \{1, 2\}, i = 1, \ldots, N_j\}$.

# Data model

- We define the GMM separation quality factor (interpreted as SNR):

$$\mathcal{S} := \left( \frac{\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|}{\sigma_1 + \sigma_2} \right)^2 = \frac{\|\boldsymbol{\mu}\|^2}{\sigma^2}.$$

  In contrast to many theoretical papers, our analysis covers SNR arbitrarily close to zero.
- The training data consists of $N_j$ labeled i.i.d. samples per class $j$, denoted by $\mathcal{D} = \{\mathbf{x}_{i,j} : j \in \{1, 2\}, i = 1, \ldots, N_j\}$.
- Without loss of generality, we denote $N_1 = N$ and $N_2 = \gamma N$ for some $\gamma \in (0, 1]$.

# The Classifier

- In the considered setting, the optimal Bayes classifier reads:

$$c_{opt}(\mathbf{x}) = \underset{j\in\{1,2\}}{\arg\max}\ \pi_j p_{x|y}(\mathbf{x}|j) = \underset{j\in\{1,2\}}{\arg\min} \|\mathbf{x} - \boldsymbol{\mu}_j\| .$$

# The Classifier

- In the considered setting, the optimal Bayes classifier reads:

$$c_{opt}(\mathbf{x}) = \underset{j\in\{1,2\}}{\arg\max}\ \pi_j p_{x|y}(\mathbf{x}|j) = \underset{j\in\{1,2\}}{\arg\min} \|\mathbf{x} - \boldsymbol{\mu}_j\| .$$

- In practice, the data distribution is unknown, so a classifier must estimate the class means $\{\boldsymbol{\mu}_i\}$ from the training set.

$$\boxed{\widehat{c}(\mathbf{x};\mathcal{D}) = \underset{j\in\{1,2\}}{\arg\min} \|\mathbf{x} - \widehat{\boldsymbol{\mu}}_j(\mathcal{D})\|},$$

where $\widehat{\boldsymbol{\mu}}_j(\mathcal{D}) = \frac{1}{N_j}\sum_{i=1}^{N_j} \mathbf{x}_{i,j}$ is the maximum likelihood estimate of $\boldsymbol{\mu}_j$.

# The Classifier

- In the considered setting, the optimal Bayes classifier reads:

$$c_{opt}(\mathbf{x}) = \underset{j\in\{1,2\}}{\arg\max}\ \pi_j p_{x|y}(\mathbf{x}|j) = \underset{j\in\{1,2\}}{\arg\min} \|\mathbf{x} - \boldsymbol{\mu}_j\| .$$

- In practice, the data distribution is unknown, so a classifier must estimate the class means $\{\boldsymbol{\mu}_i\}$ from the training set.

$$\boxed{\widehat{c}(\mathbf{x};\mathcal{D}) = \underset{j\in\{1,2\}}{\arg\min} \|\mathbf{x} - \widehat{\boldsymbol{\mu}}_j(\mathcal{D})\|},$$

  where $\widehat{\boldsymbol{\mu}}_j(\mathcal{D}) = \frac{1}{N_j}\sum_{i=1}^{N_j} \mathbf{x}_{i,j}$ is the maximum likelihood estimate of $\boldsymbol{\mu}_j$.

- $\widehat{\boldsymbol{\mu}}_j \sim \mathcal{N}\left(\boldsymbol{\mu}_j, \frac{\sigma^2}{N_j}\boldsymbol{I}_d\right)$ is an efficient estimator of $\boldsymbol{\mu}_j$.

# The Classifier

- In the considered setting, the optimal Bayes classifier reads:

$$c_{opt}(\mathbf{x}) = \underset{j\in\{1,2\}}{\arg\max}\ \pi_j p_{x|y}(\mathbf{x}|j) = \underset{j\in\{1,2\}}{\arg\min} \|\mathbf{x} - \boldsymbol{\mu}_j\| .$$

- In practice, the data distribution is unknown, so a classifier must estimate the class means $\{\boldsymbol{\mu}_i\}$ from the training set.

$$\boxed{\widehat{c}(\mathbf{x};\mathcal{D}) = \underset{j\in\{1,2\}}{\arg\min} \|\mathbf{x} - \widehat{\boldsymbol{\mu}}_j(\mathcal{D})\|},$$

  where $\widehat{\boldsymbol{\mu}}_j(\mathcal{D}) = \frac{1}{N_j}\sum_{i=1}^{N_j} \mathbf{x}_{i,j}$ is the maximum likelihood estimate of $\boldsymbol{\mu}_j$.

- $\widehat{\boldsymbol{\mu}}_j \sim \mathcal{N}\left(\boldsymbol{\mu}_j, \frac{\sigma^2}{N_j}\boldsymbol{I}_d\right)$ is an efficient estimator of $\boldsymbol{\mu}_j$.
- $\widehat{c}(\cdot)$ is structurally similar to $c_{\text{opt}}(\cdot)$ and converges to it as $N_j \to \infty$.

# Data Processing

- We study a linear dimensionality reduction to $1 \leq k < d$. Specifically, we consider

$$\mathbf{z} = \boldsymbol{A}\mathbf{x}$$

with $\boldsymbol{A} \in \mathbb{R}^{k \times d}$ that obeys

$$\boldsymbol{A}\boldsymbol{A}^{\top} = \boldsymbol{I}_k, \qquad \|\boldsymbol{A}\boldsymbol{\mu}\| = \|\boldsymbol{\mu}\| .$$

# Data Processing

- We study a linear dimensionality reduction to $1 \leq k < d$. Specifically, we consider

$$\mathbf{z} = \boldsymbol{A}\mathbf{x}$$

with $\boldsymbol{A} \in \mathbb{R}^{k \times d}$ that obeys

$$\boldsymbol{A}\boldsymbol{A}^{\top} = \boldsymbol{I}_k, \qquad \|\boldsymbol{A}\boldsymbol{\mu}\| = \|\boldsymbol{\mu}\| .$$

- We provide a constructive proof that shows how such $\boldsymbol{A}$ can be learned from unlabeled data without prior knowledge of $\boldsymbol{\mu}$.

# Theorem 2: The probability of error before the processing

## Theorem 2 (The probability of error before the processing)

*With approximation accuracy* $\mathcal{O}(1/\sqrt{d})$ *we have* $p_{\mathbf{x}}(\text{error}) \approx \hat{p}_{\mathbf{x}}(\text{error}) = \hat{p}(\mathcal{S}, N, \gamma, d)$, *where*

$$\hat{p}(\mathcal{S}, N, \gamma, d) := \frac{1}{2} \cdot \mathcal{Q}\left( \frac{\sqrt{\mathcal{S}} + \frac{1}{4N} \cdot \frac{1-\gamma}{\gamma} \cdot \frac{d}{\sqrt{\mathcal{S}}}}{\sqrt{\frac{1}{4N} \cdot \frac{1+\gamma}{\gamma} \cdot \frac{d}{\mathcal{S}} + \frac{1}{8N^2} \cdot \frac{1+\gamma^2}{\gamma^2} \cdot \frac{d}{\mathcal{S}} + \frac{1}{\gamma N} + 1}} \right)$$
$$+ \frac{1}{2} \cdot \mathcal{Q}\left( \frac{\sqrt{\mathcal{S}} - \frac{1}{4N} \cdot \frac{1-\gamma}{\gamma} \cdot \frac{d}{\sqrt{\mathcal{S}}}}{\sqrt{\frac{1}{4N} \cdot \frac{1+\gamma}{\gamma} \cdot \frac{d}{\mathcal{S}} + \frac{1}{8N^2} \cdot \frac{1+\gamma^2}{\gamma^2} \cdot \frac{d}{\mathcal{S}} + \frac{1}{N} + 1}} \right).$$

# Theorem 3: The existence and learnability of the processing

## Theorem 3 (The existence and learnability of the processing)

*For all $1 \leq k < d$, there exists a dimension-reducing matrix $\boldsymbol{A} \in \mathbb{R}^{k \times d}$ with the following properties:*

$$\boldsymbol{A}\boldsymbol{A}^{\top} = \boldsymbol{I}_k, \qquad \|\boldsymbol{A}\boldsymbol{\mu}\| = \|\boldsymbol{\mu}\| .$$

*Furthermore, given sufficiently many unlabeled samples, such a matrix can be learned to arbitrary accuracy.*

# Theorem 4: The probability of error on the processed data

## Theorem 4 (The probability of error on the processed data)

*With approximation accuracy* $\mathcal{O}(1/\sqrt{k})$ *we have* $p_{\mathbf{z}}(\text{error}) \approx \hat{p}_{\mathbf{z}}(\text{error}) = \hat{p}\,(\mathcal{S}, N, \gamma, k)$, *where*

$$\hat{p}(\mathcal{S}, N, \gamma, k) := \frac{1}{2} \cdot \mathcal{Q}\left( \frac{\sqrt{\mathcal{S}} + \frac{1}{4N} \cdot \frac{1-\gamma}{\gamma} \cdot \frac{k}{\sqrt{\mathcal{S}}}}{\sqrt{\frac{1}{4N} \cdot \frac{1+\gamma}{\gamma} \cdot \frac{k}{\mathcal{S}} + \frac{1}{8N^2} \cdot \frac{1+\gamma^2}{\gamma^2} \cdot \frac{k}{\mathcal{S}} + \frac{1}{\gamma N} + 1}} \right)$$
$$+ \frac{1}{2} \cdot \mathcal{Q}\left( \frac{\sqrt{\mathcal{S}} - \frac{1}{4N} \cdot \frac{1-\gamma}{\gamma} \cdot \frac{k}{\sqrt{\mathcal{S}}}}{\sqrt{\frac{1}{4N} \cdot \frac{1+\gamma}{\gamma} \cdot \frac{k}{\mathcal{S}} + \frac{1}{8N^2} \cdot \frac{1+\gamma^2}{\gamma^2} \cdot \frac{k}{\mathcal{S}} + \frac{1}{N} + 1}} \right).$$

- Note that this is the same formula as before processing, except that we have replaced $d$ with $k$.

# Theorem 4: The probability of error on the processed data

## Theorem 4 (The probability of error on the processed data)

*With approximation accuracy* $\mathcal{O}(1/\sqrt{k})$ *we have* $p_{\mathbf{z}}(\text{error}) \approx \hat{p}_{\mathbf{z}}(\text{error}) = \hat{p}\,(\mathcal{S}, N, \gamma, k)$, *where*

$$\hat{p}(\mathcal{S}, N, \gamma, k) := \frac{1}{2} \cdot \mathcal{Q}\left( \frac{\sqrt{\mathcal{S}} + \frac{1}{4N} \cdot \frac{1-\gamma}{\gamma} \cdot \frac{k}{\sqrt{\mathcal{S}}}}{\sqrt{\frac{1}{4N} \cdot \frac{1+\gamma}{\gamma} \cdot \frac{k}{\mathcal{S}} + \frac{1}{8N^2} \cdot \frac{1+\gamma^2}{\gamma^2} \cdot \frac{k}{\mathcal{S}} + \frac{1}{\gamma N} + 1}} \right)$$
$$+ \frac{1}{2} \cdot \mathcal{Q}\left( \frac{\sqrt{\mathcal{S}} - \frac{1}{4N} \cdot \frac{1-\gamma}{\gamma} \cdot \frac{k}{\sqrt{\mathcal{S}}}}{\sqrt{\frac{1}{4N} \cdot \frac{1+\gamma}{\gamma} \cdot \frac{k}{\mathcal{S}} + \frac{1}{8N^2} \cdot \frac{1+\gamma^2}{\gamma^2} \cdot \frac{k}{\mathcal{S}} + \frac{1}{N} + 1}} \right).$$

- Note that this is the same formula as before processing, except that we have replaced $d$ with $k$.
- This follows because $\mathbf{z}$ follows a GMM with the same parameters $\gamma$ and $\mathcal{S}$, except that its dimensionality is reduced to $k$.

# Theorem 5: Performance gain under balanced training data

## Theorem 5 (Performance gain under balanced training data)

*For $\gamma = 1$, and for all $\mathcal{S} > 0$, $1 \leq k < d$, and $N \in \mathbb{N}$, we have*

$$\hat{p}_{\mathbf{x}}(\text{error}) > \hat{p}_{\mathbf{z}}(\text{error}).$$

# Theorem 6: Performance gain under imbalanced training data

**Theorem 6 (Performance gain under imbalanced training data)**

*Let* $0 < \gamma < 1,\ 0 < \mathcal{S} \leq 1,\ 1 \leq k < d$. *If* $N \geq \dfrac{\gamma^2 - 4\gamma + 1}{2\gamma(1+\gamma)}$, *then we have*

$$\hat{p}_{\mathbf{x}}(\text{error}) > \hat{p}_{\mathbf{z}}(\text{error}).$$

Note that these assumptions are reasonable and still encompass the interesting case of low SNR and a reasonable number of training samples.

# Defining the efficiency of the processing

So far, we have only considered the relation between $\hat{p}_{\mathbf{x}}(\text{error})$ and $\hat{p}_{\mathbf{z}}(\text{error})$. Let us now discuss the margin between them, which reflects the efficiency of the processing.

## Definition 1

*We define the theoretical efficiency of the processing as*

$$\eta \coloneqq \left( \frac{\hat{p}_{\mathbf{x}}(\text{error}) - \hat{p}_{\mathbf{z}}(\text{error})}{\hat{p}_{\mathbf{x}}(\text{error})} \right) \cdot 100.$$

# Theorem 7: Analysis of the asymptotic efficiency

## Theorem 7 (Analysis of the asymptotic efficiency)

*Let $\mathcal{S} > 0,\ 1 \leq k < d,\ 0 < \gamma \leq 1$. Denote by $N_T = (1+\gamma)\,N$ the total number of training samples. With approximation accuracy $\mathcal{O}(1/N_T^2)$, we have*

$$\eta \approx \frac{25}{2\sqrt{2\pi}} \cdot \frac{\exp\left(-\frac{\mathcal{S}}{2}\right)}{\sqrt{\mathcal{S}} \cdot \mathcal{Q}\left(\sqrt{\mathcal{S}}\right)} \cdot \left(3 + 2\gamma + \frac{1}{\gamma}\right) \cdot (d-k) \cdot \frac{1}{N_T}.$$

In particular, for $N_T \gg 1$:

- The efficiency increases when $d - k$ increases.

# Theorem 7: Analysis of the asymptotic efficiency

## Theorem 7 (Analysis of the asymptotic efficiency)

*Let $\mathcal{S} > 0,\ 1 \le k < d,\ 0 < \gamma \le 1$. Denote by $N_T = (1+\gamma)\,N$ the total number of training samples. With approximation accuracy $\mathcal{O}(1/N_T^2)$, we have*

$$\eta \approx \frac{25}{2\sqrt{2\pi}} \cdot \frac{\exp\left(-\frac{\mathcal{S}}{2}\right)}{\sqrt{\mathcal{S}} \cdot \mathcal{Q}\left(\sqrt{\mathcal{S}}\right)} \cdot \left(3 + 2\gamma + \frac{1}{\gamma}\right) \cdot (d-k) \cdot \frac{1}{N_T}.$$

In particular, for $N_T \gg 1$:

- The efficiency increases when $d - k$ increases.
- The efficiency increases when $\gamma$ decreases within $0 < \gamma \le 1/\sqrt{2}$.

# Theorem 7: Analysis of the asymptotic efficiency

## Theorem 7 (Analysis of the asymptotic efficiency)

*Let $\mathcal{S} > 0,\ 1 \leq k < d,\ 0 < \gamma \leq 1$. Denote by $N_T = (1+\gamma)\,N$ the total number of training samples. With approximation accuracy $\mathcal{O}(1/N_T^2)$, we have*

$$\eta \approx \frac{25}{2\sqrt{2\pi}} \cdot \frac{\exp\left(-\frac{\mathcal{S}}{2}\right)}{\sqrt{\mathcal{S}} \cdot \mathcal{Q}\left(\sqrt{\mathcal{S}}\right)} \cdot \left(3 + 2\gamma + \frac{1}{\gamma}\right) \cdot (d-k) \cdot \frac{1}{N_T}.$$

In particular, for $N_T \gg 1$:

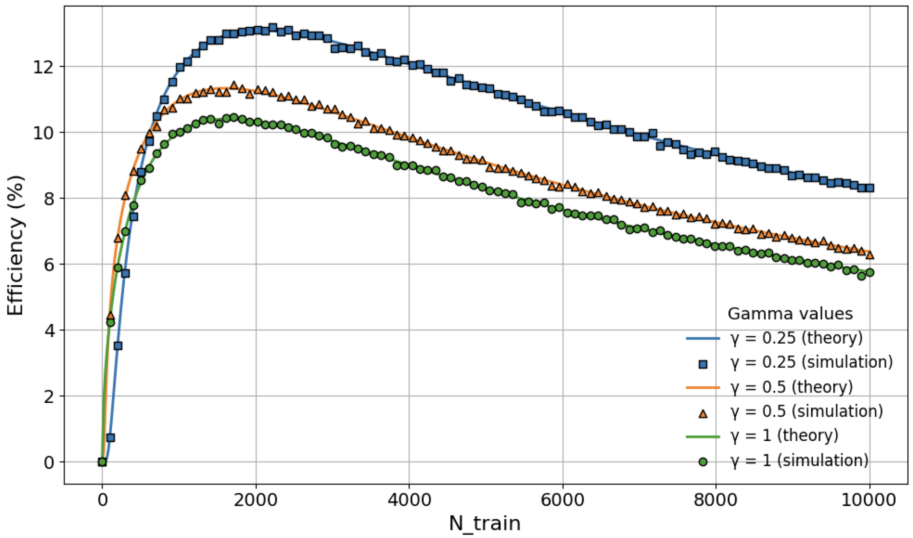
- The efficiency increases when $d - k$ increases.
- The efficiency increases when $\gamma$ decreases within $0 < \gamma \leq 1/\sqrt{2}$.
- The efficiency decreases when $\mathcal{S}$ increases or $N_T$ increases.

# Theorem 8: Analysis of the maximal efficiency

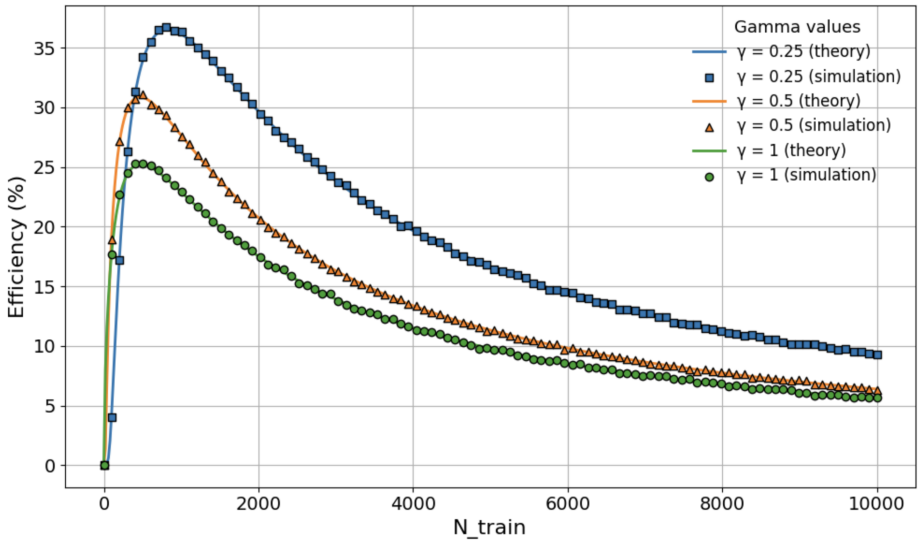
## Theorem 8 (Analysis of the maximal efficiency)

*Fix* $\gamma = 1$, *and let* $\mathcal{S} > 0, 1 \leq k < d$. *Consider the efficiency* $\eta = \eta(N)$ *as a function of continuous* $N \in \mathbb{R}_+$. *We have that the maximal efficiency* $\eta_{\max} = \max_{N \geq 0} \eta(N)$ *increases as a function of* $\mathcal{S}$.

# Empirical Verification



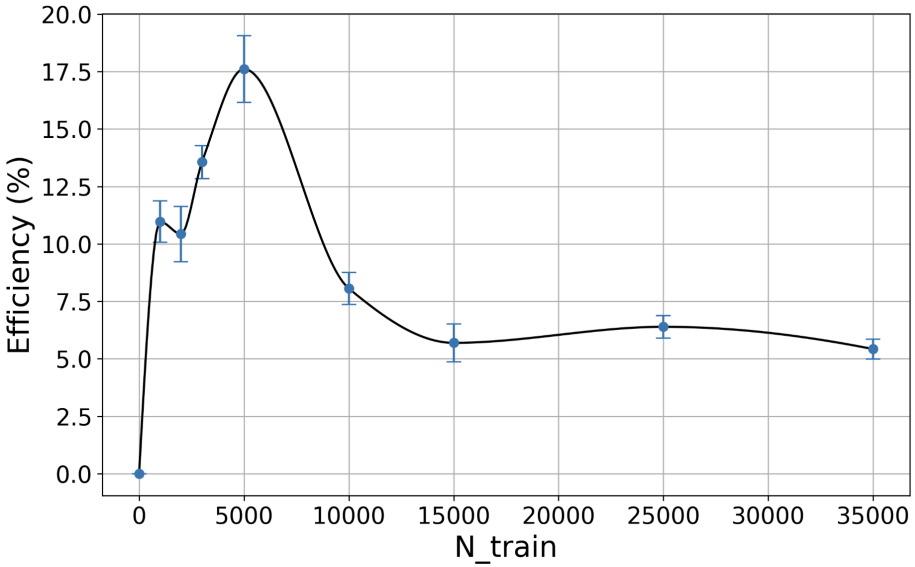

$\mathcal{S} = 0.75^2$



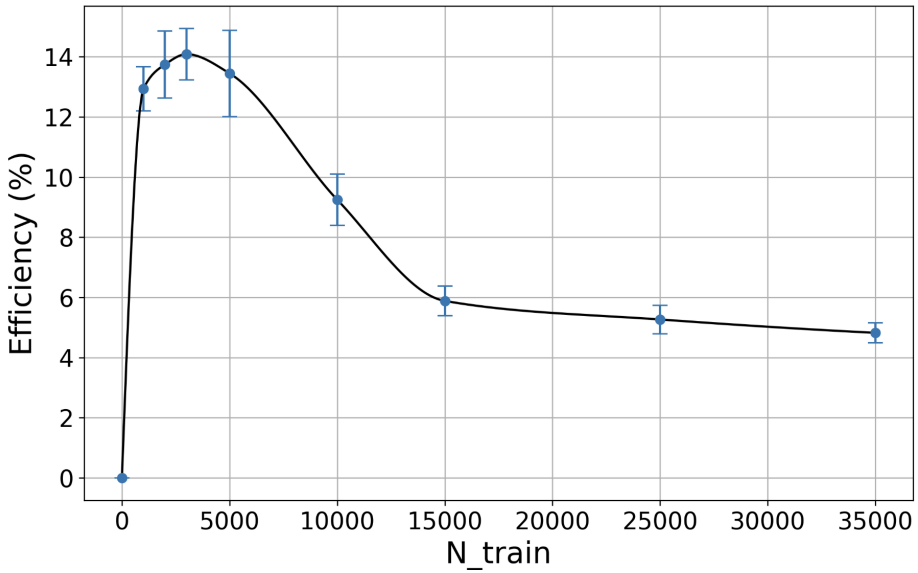

$\mathcal{S} = 1.5^2$

Figure: The theoretical setup. Efficiency of the data processing procedure versus the number of training samples $N_{\text{train}}$, for various values of the training imbalance factor, $\gamma$, and the SNR, $\mathcal{S}$.

# Noisy CIFAR-10 and pre-classification denoising



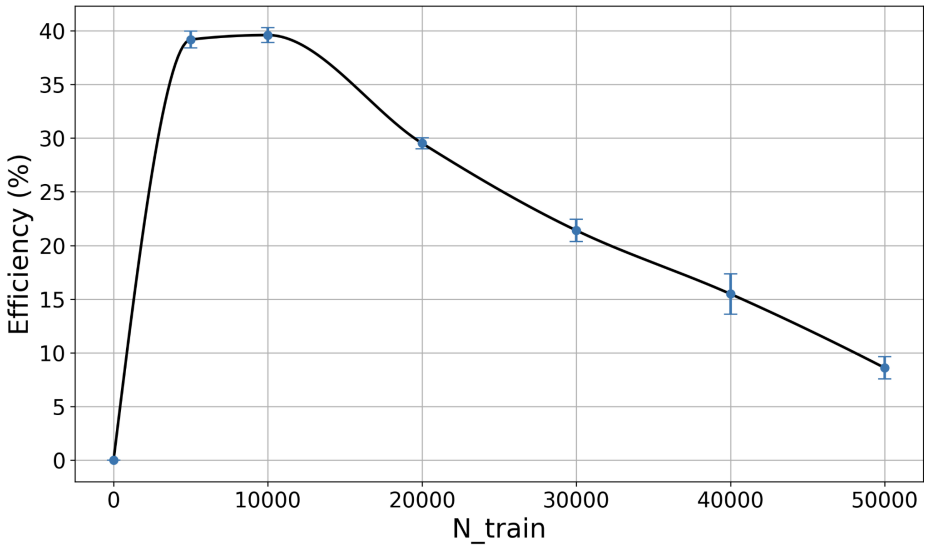

$\sigma = 0.25, \gamma = 1$



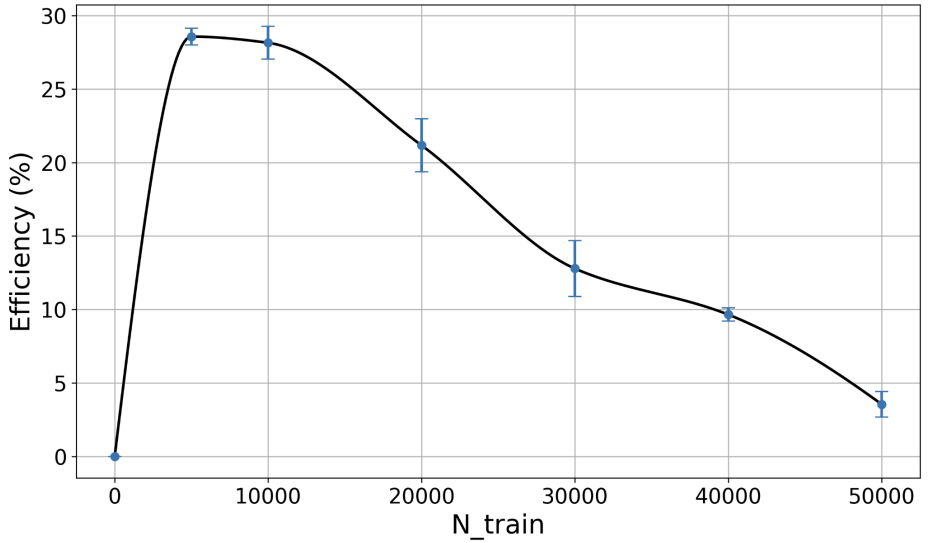

$\sigma = 0.4, \gamma = 1$

Figure: Noisy CIFAR-10 and pre-classification denoising. Efficiency versus $N_{\text{train}}$.

# Noisy Mini-ImageNet and pre-classification encoding



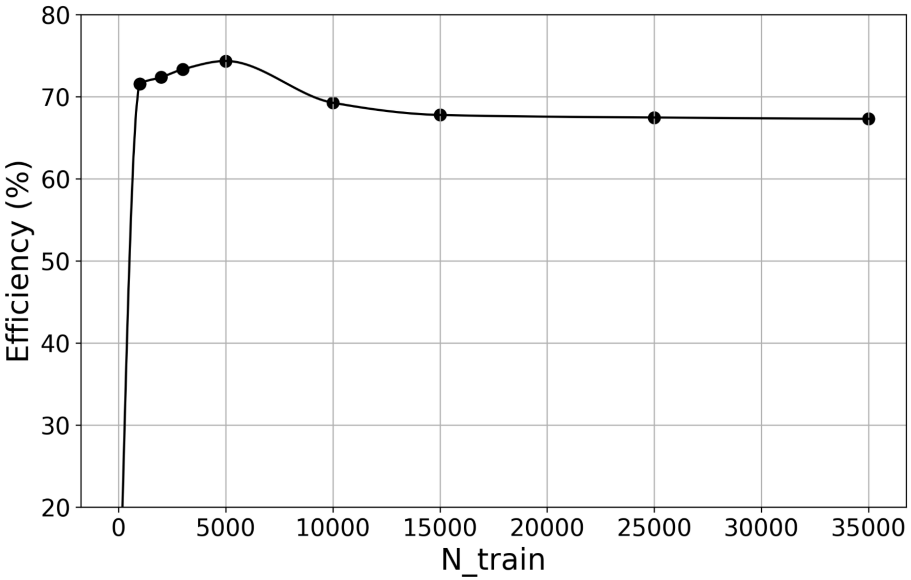

$$\sigma = \frac{50}{255}, \gamma = 1$$



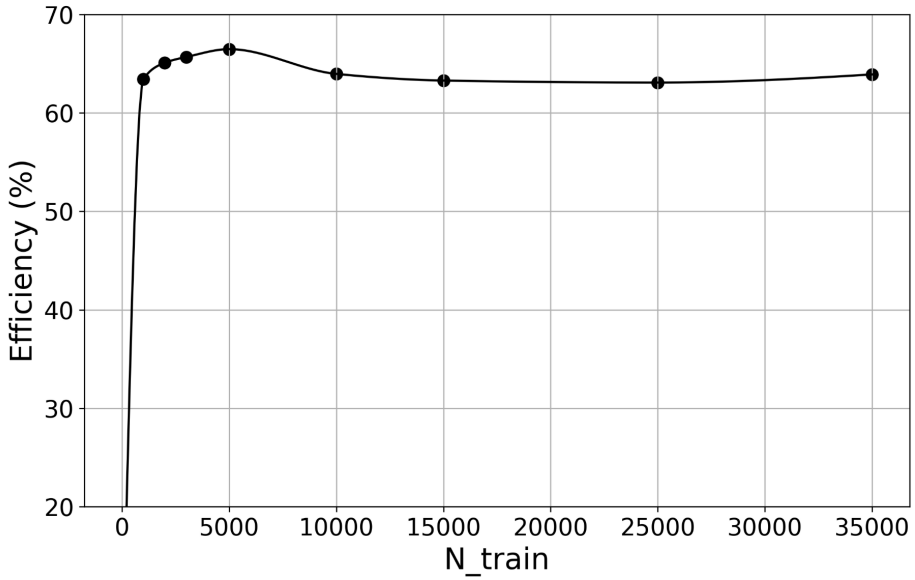

$$\sigma = \frac{100}{255}, \gamma = 1$$

Figure: Noisy Mini-ImageNet and pre-classification encoding. Efficiency versus $N_{\text{train}}$.

# Noisy CIFAR-10 and pre-classification encoding




$\sigma = 0.25, \gamma = 1$




$\sigma = 0.4, \gamma = 1$

Figure: Noisy CIFAR-10 and pre-classification encoding. Efficiency versus $N_{\text{train}}$.

# Practical experiments discussion

- To sum up, the trends observed in practical deep learning experiments are aligned with our theory: 1) a similar non-monotonicity of the curve while remaining positive, and 2) the maximal efficiency increases with the SNR.

# Practical experiments discussion

- To sum up, the trends observed in practical deep learning experiments are aligned with our theory: 1) a similar non-monotonicity of the curve while remaining positive, and 2) the maximal efficiency increases with the SNR.
- Main message to practitioners: when labeled samples are scarce, data processing can be especially advantageous for 'high quality' data.

# Practical experiments discussion

- To sum up, the trends observed in practical deep learning experiments are aligned with our theory: 1) a similar non-monotonicity of the curve while remaining positive, and 2) the maximal efficiency increases with the SNR.
- Main message to practitioners: when labeled samples are scarce, data processing can be especially advantageous for 'high quality' data.
- Notice that the efficiency gains are much higher when using encoding as the low-level task rather than denoising. This suggests that for the classification task, encoding may be a more effective low-level task.

# Practical experiments discussion

- To sum up, the trends observed in practical deep learning experiments are aligned with our theory: 1) a similar non-monotonicity of the curve while remaining positive, and 2) the maximal efficiency increases with the SNR.
- Main message to practitioners: when labeled samples are scarce, data processing can be especially advantageous for 'high quality' data.
- Notice that the efficiency gains are much higher when using encoding as the low-level task rather than denoising. This suggests that for the classification task, encoding may be a more effective low-level task.
- However, we believe that this may not be the case for other high-level tasks, which may require preserving spatial information in the image (e.g., object detection).

# Conclusion

- Low-level preprocessing can improve classification accuracy, even when using a classifier that converges to the Bayes-optimal classifier as the sample size increases.

# Conclusion

- Low-level preprocessing can improve classification accuracy, even when using a classifier that converges to the Bayes-optimal classifier as the sample size increases.
- The gain from preprocessing depends on the sample size, SNR, and class imbalance. Moreover, the maximal gain increases as the SNR increases.

# Conclusion

- Low-level preprocessing can improve classification accuracy, even when using a classifier that converges to the Bayes-optimal classifier as the sample size increases.
- The gain from preprocessing depends on the sample size, SNR, and class imbalance. Moreover, the maximal gain increases as the SNR increases.
- Empirical trends observed in image denoising and encoding are consistent with the theoretical predictions.

# Conclusion

- Low-level preprocessing can improve classification accuracy, even when using a classifier that converges to the Bayes-optimal classifier as the sample size increases.
- The gain from preprocessing depends on the sample size, SNR, and class imbalance. Moreover, the maximal gain increases as the SNR increases.
- Empirical trends observed in image denoising and encoding are consistent with the theoretical predictions.
- In out-of-distribution scenarios, preprocessing that reduces the gap between the training and test sets could provide even greater benefits.

# Conclusion

- Low-level preprocessing can improve classification accuracy, even when using a classifier that converges to the Bayes-optimal classifier as the sample size increases.
- The gain from preprocessing depends on the sample size, SNR, and class imbalance. Moreover, the maximal gain increases as the SNR increases.
- Empirical trends observed in image denoising and encoding are consistent with the theoretical predictions.
- In out-of-distribution scenarios, preprocessing that reduces the gap between the training and test sets could provide even greater benefits.
- As future work, one may further analyze the effect of $\gamma$ on efficiency and extend the study beyond classification to other high-level tasks.

# Conclusion

- Low-level preprocessing can improve classification accuracy, even when using a classifier that converges to the Bayes-optimal classifier as the sample size increases.
- The gain from preprocessing depends on the sample size, SNR, and class imbalance. Moreover, the maximal gain increases as the SNR increases.
- Empirical trends observed in image denoising and encoding are consistent with the theoretical predictions.
- In out-of-distribution scenarios, preprocessing that reduces the gap between the training and test sets could provide even greater benefits.
- As future work, one may further analyze the effect of $\gamma$ on efficiency and extend the study beyond classification to other high-level tasks.
- Finally, one may attempt to characterize the optimal unsupervised low-level processing, given a high-level task.

# References I

Haris, Muhammad, Greg Shakhnarovich, and Norimichi Ukita (2021). "Task-driven super resolution: Object detection in low-resolution images". In: *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part V 28*. Springer, pp. 387–395.

Kothapalli, Vignesh and Tom Tirer (2025). "Can Kernel Methods Explain How the Data Affects Neural Collapse?" In: *Transactions on Machine Learning Research*. ISSN: 2835-8856.

Liu, Dong, Haochen Zhang, and Zhiwei Xiong (2019). "On the classification-distortion-perception tradeoff". In: *Advances in Neural Information Processing Systems* 32.