LAMARR

INSTITUTE FOR MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Generative Adversarial Networks and Active Learning

Dr. Amal Saadallah

Partner institutions:

technische universität dortmund

Fraunhofer IAIS

UNIVERSITÄT BONN

Fraunhofer IML

Institutionally funded by:

Bundesministerium für Bildung und Forschung

Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen

The Lamarr Institute

Partners & Locations

2

Universities

TU Dortmund University
University of Bonn

2

Fraunhofer Institutes

Fraunhofer IAIS
Fraunhofer IML

3

Locations

Bonn
Dortmund
Sankt Augustin

38

Principal Investigators

The Lamarr Institute

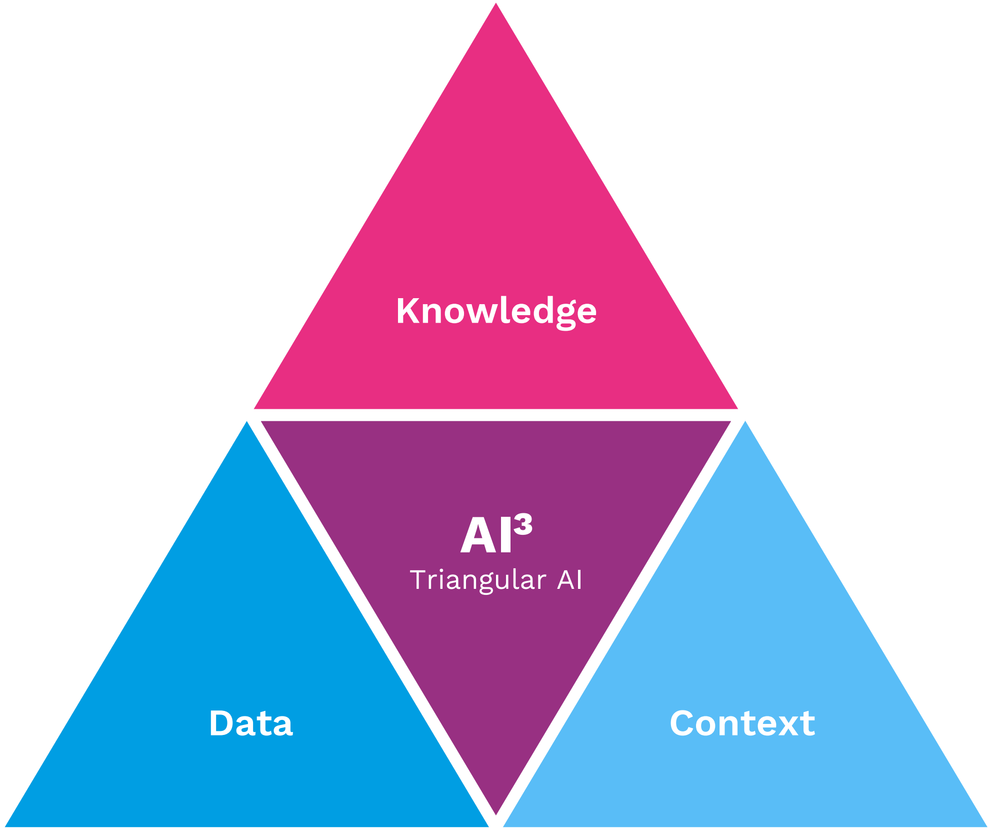
Research Areas and Mission

Core Research Areas

- Hybrid ML
- Resource-aware ML
- Human-centered AI Systems
- Trustworthy AI
- Embodied AI

Interdisciplinary Research Areas

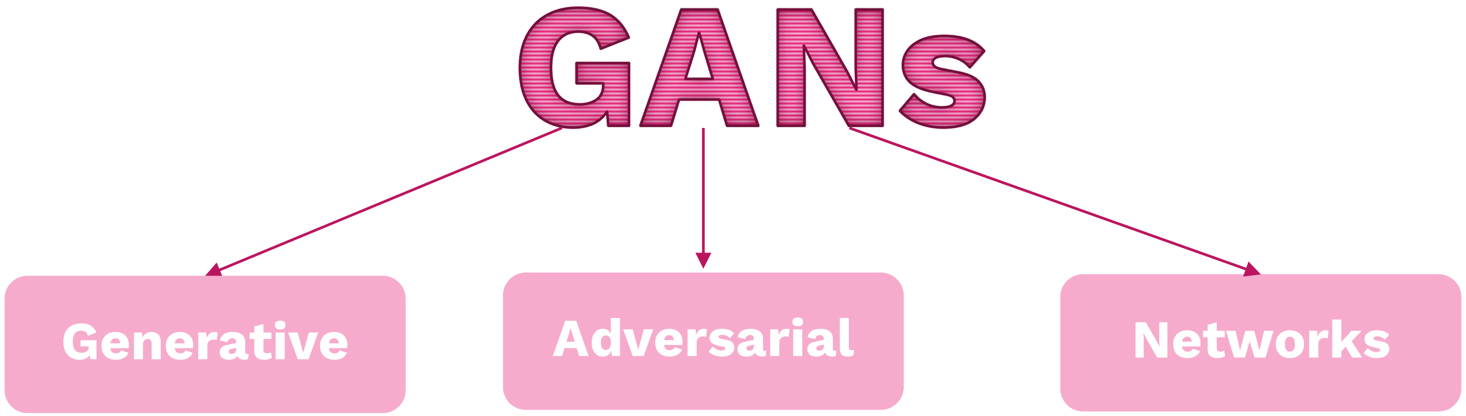
- Planning & Logistics
- Physics
- Industry & Production
- Life Sciences
- Natural Language Processing (NLP)



Outline

- Generative Learning and Adversarial Training
- Active Learning
- Generative Adversarial Active Learning
- Concluding Remarks

Generative Learning and Adversarial Training



Generative Learning and Adversarial Training

Generative Learning

Discriminative Learning ≠ Generative Learning

Generative Learning

Discriminative Learning $\neq$ **Generative Learning**

The goal is to model the conditional probability distribution $P(Y|X)$ directly

- ➢ Predict labels given input features.

Generative Learning

Discriminative Learning $\neq$ **Generative Learning**

The goal is to model the conditional probability distribution $P(Y|X)$ directly

- ➢ Predict labels given input features.

The goal is to model the joint probability distribution $P(X,Y)$ of input features X and corresponding labels Y.

- ➢ Learn the underlying data distribution
- ➢ Capture the dependencies between input features and labels.
- ➢ Generate new samples from that distribution

Generative Learning and Adversarial Training

Generative Learning

Role of Deep Learning in Generative Modelling:

- **Representation Learning:**
  - ❑ Learning rich hierarchical representations of data
  - ❑ Capturing patterns and structures in the data
  - ❑ Modelling complex data distributions more effectively.

Generative Learning

Role of Deep Learning in Generative Modelling:

- **Representation Learning:**
  - ❑ Learning rich hierarchical representations of data
  - ❑ Capturing intricate patterns and structures in the data
  - ❑ Modelling complex data distributions more effectively.

- **Architectural Flexibility:**
  - ❑ A wide range of neural network architectures, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers.
  - ❑ Suitable to different types of data and modelling objectives.

Generative Learning and Adversarial Training

Generative Learning

Role of Deep Learning in Generative Modelling:

- **Representation Learning:**
  - ❑ Learning rich hierarchical representations of data
  - ❑ Capturing intricate patterns and structures in the data
  - ❑ Modelling complex data distributions more effectively.

- **Architectural Flexibility:**
  - ❑ A wide range of neural network architectures, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers.
  - ❑ Suitable to different types of data and modelling objectives.

- **Scalability:**
  - ❑ Advancements of deep learning frameworks and hardware for large-scale generative models on vast amounts of data efficiently.
  - ❑ More powerful generative models that can capture diverse and high-dimensional data distributions.

Generative Learning and Adversarial Training

Generative Learning

Popular types of generative learning include:

- **Variational Autoencoders (VAEs):**
  - ❑ Combine DNNs with variational inference to learn probabilistic latent representations of data.
  - ❑ Generate new samples by sampling from the learned latent space

Generative Learning and Adversarial Training

Generative Learning

Popular types of generative learning include:

- **Variational Autoencoders (VAEs):**
  - ❑ Combine DNNs with variational inference to learn probabilistic latent representations of data.
  - ❑ Generate new samples by sampling from the learned latent space
- **Transformer Models:**
  - ❑ Developed for natural language processing tasks
  - ❑ Leverage self-attention mechanisms to capture long-range dependencies in data sequences, making them well-suited for tasks such as text generation, image generation, and sequence-to-sequence modelling.

Generative Learning and Adversarial Training

Generative Learning

Popular types of generative learning include:

- **Variational Autoencoders (VAEs):**
  - ❑ Combine DNNs with variational inference to learn probabilistic latent representations of data.
  - ❑ Generate new samples by sampling from the learned latent space
- **Transformer Models:**
  - ❑ Developed for natural language processing tasks
  - ❑ Leverage self-attention mechanisms to capture long-range dependencies in data sequences, making them well-suited for tasks such as text generation, image generation, and sequence-to-sequence modelling.
- **Autoregressive Models:**
  - ❑ Model the conditional distribution of each feature given previous features in the sequence.
  - ❑ Generate data sequentially, one feature at a time.

Generative Learning and Adversarial Training

Generative Learning

Popular types of generative learning include:

- **Flow-Based Models:**
  - ❑ Parameterize complex data distributions through a series of invertible transformations.
  - ❑ learn an explicit probability density function of the data, enabling efficient sampling and likelihood estimation.

Generative Learning and Adversarial Training

Generative Learning

Popular types of generative learning include:

- **Flow-Based Models:**
  - ❑ Parameterize complex data distributions through a series of invertible transformations.
  - ❑ learn an explicit probability density function of the data, enabling efficient sampling and likelihood estimation.
- **Generative Moment Matching Networks (GMMNs):**
  - ❑ Learn to match the moments of the data distribution using a feedforward neural network.
  - ❑ Optimize a similarity measure between the generated and real data distributions.

Generative Learning and Adversarial Training

Generative Learning

Popular types of generative learning include:

- **Flow-Based Models:**
  - ❑ Parameterize complex data distributions through a series of invertible transformations.
  - ❑ learn an explicit probability density function of the data, enabling efficient sampling and likelihood estimation.
- **Generative Moment Matching Networks (GMMNs):**
  - ❑ Learn to match the moments of the data distribution using a feedforward neural network.
  - ❑ Optimize a similarity measure between the generated and real data distributions.
- **Probabilistic Graphical Models (PGMs):**
  - ❑ Represent the joint distribution of random variables using graphical structures.
  - ❑ Provide a principled framework for modelling complex dependencies between variables.

Generative Learning and Adversarial Training

Generative Learning

Popular types of generative learning include:

- **Flow-Based Models:**
  - ❑ Parameterize complex data distributions through a series of invertible transformations.
  - ❑ learn an explicit probability density function of the data, enabling efficient sampling and likelihood estimation.
- **Generative Moment Matching Networks (GMMNs):**
  - ❑ Learn to match the moments of the data distribution using a feedforward neural network.
  - ❑ Optimize a similarity measure between the generated and real data distributions.
- **Probabilistic Graphical Models (PGMs):**
  - ❑ Represent the joint distribution of random variables using graphical structures.
  - ❑ Provide a principled framework for modelling complex dependencies between variables.
- **Generative Adversarial Networks (GANs):**
  - ❑ Consists of two neural networks, a generator and a discriminator, which are trained adversarially to generate realistic samples.
  - ❑ Demonstrated impressive performance in generating images, audio, text, and other types of data.

Generative Learning and Adversarial Training

Generative Learning

Advantages of Generative Learning

- **Data Generation:**
  - ❑ Generate new data samples that resemble the original training data.
  - ❑ Image synthesis, text generation, and data augmentation, etc.

Generative Learning

Advantages of Generative Learning

- **Data Generation:**
  - ❑ Generate new data samples that resemble the original training data.
  - ❑ Image synthesis, text generation, and data augmentation, etc.
- **Unsupervised Learning:**
  - ❑ Learn representations of data without requiring labelled training examples
  - ❑ Capture underlying patterns and structures in the data without explicit supervision.

Generative Learning and Adversarial Training

Generative Learning

Advantages of Generative Learning

- **Data Generation:**
  - ❑ Generate new data samples that resemble the original training data.
  - ❑ Image synthesis, text generation, and data augmentation, etc.
- **Unsupervised Learning:**
  - ❑ Learn representations of data without requiring labelled training examples
  - ❑ Capture underlying patterns and structures in the data without explicit supervision.
- **Anomaly Detection:**
  - ❑ Learning the normal data distribution and identifying instances that deviate significantly from this distribution.
  - ❑ Detecting rare or abnormal data points in various domains.

Generative Learning

Advantages of Generative Learning

- **Data Generation:**
  - ❑ Generate new data samples that resemble the original training data.
  - ❑ Image synthesis, text generation, and data augmentation, etc.
- **Unsupervised Learning:**
  - ❑ Learn representations of data without requiring labelled training examples
  - ❑ Capture underlying patterns and structures in the data without explicit supervision.
- **Anomaly Detection:**
  - ❑ Learning the normal data distribution and identifying instances that deviate significantly from this distribution.
  - ❑ Detecting rare or abnormal data points in various domains.
- **Representation Learning:**
  - ❑ Capture important features and characteristics.
  - ❑ Improve downstream tasks such as classification, clustering, and retrieval.

Generative Learning and Adversarial Training

Generative Learning

Advantages of Generative Learning

- **Data Imputation:**
  - ❑ Fill in missing or corrupted data values.

Generative Learning and Adversarial Training

Generative Learning

Advantages of Generative Learning

- **Data Imputation:**
  - ❑ Fill in missing or corrupted data values.
- **Domain Adaptation:**
  - ❑ Adapt to new domains by capturing the underlying data distribution and generating data samples
  - ❑ Transfer learning

Generative Learning and Adversarial Training

Generative Learning

Advantages of Generative Learning

- **Data Imputation:**
  - ❑ Fill in missing or corrupted data values.
- **Domain Adaptation:**
  - ❑ Adapt to new domains by capturing the underlying data distribution and generating data samples
  - ❑ Transfer learning
- **Parameterizing Complex Distributions:**
  - ❑ Parameterize complex data distributions through a series of invertible transformations.
  - ❑ Allow to model highly non-linear and multi-modal distributions.

Generative Learning and Adversarial Training

Adversarial Training

- Improve the robustness of machine learning models, particularly neural networks, against **adversarial examples**.

Xu, Xiaojun, et al. "Can you fool ai with adversarial examples on a visual turing test." arXiv preprint arXiv:1709.08693 3 (2017).

Generative Learning and Adversarial Training

Adversarial Training

- Improve the robustness of machine learning models, particularly neural networks, against **adversarial examples**.

↓

- ✓ Small Changes.
- ✓ Most often imperceptible changes to humans.
- ✓ Cause a model to make incorrect predictions.

Generative Learning and Adversarial Training

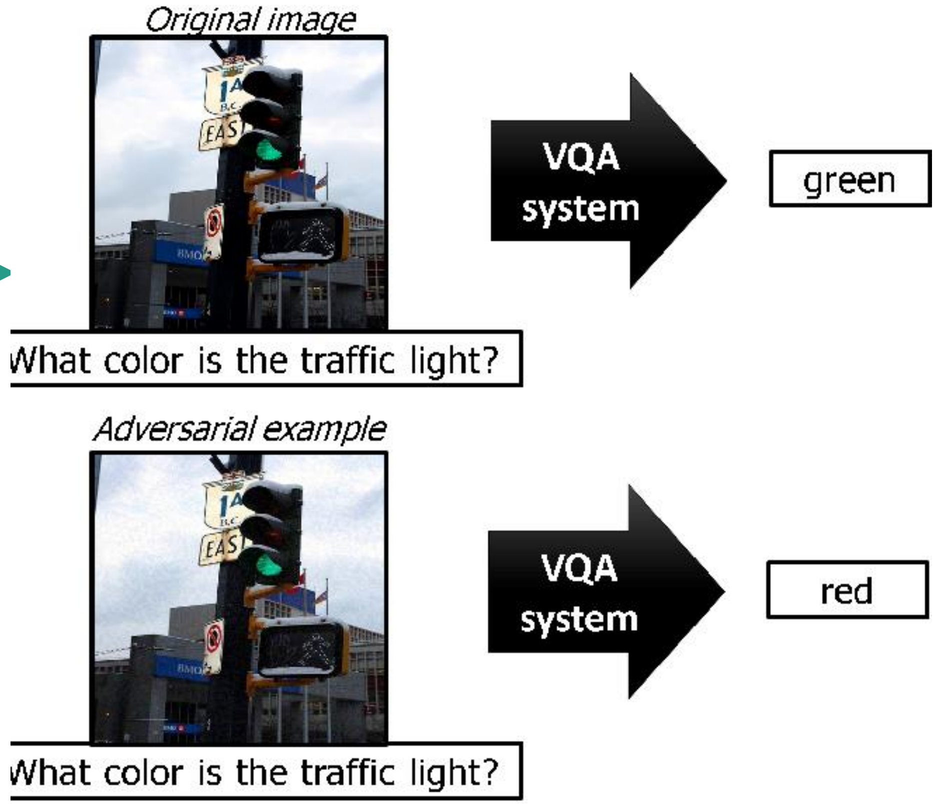
Adversarial Training

- Improve the robustness of machine learning models, particularly neural networks, against **adversarial examples**.

✓ Small Changes.

✓ Most often imperceptible changes to humans.

✓ Cause a model to make incorrect predictions.



Xu, Xiaojun, et al. "Can you fool ai with adversarial examples on a visual turing test." arXiv preprint arXiv:1709.08693 3 (2017).

Generative Learning and Adversarial Training

Adversarial Training

Generating Adversarial Examples:

- **Fast Gradient Sign Method (FGSM):**

Perturbs the input x in the direction of the gradient of the loss with respect to the input

$$x' = x + \epsilon \cdot \mathrm{sign}(\nabla_x J(\theta, x, y))$$

where x' is the adversarial example, ϵ is the perturbation magnitude, J is the loss function, θ represents the model parameters, and y is the true label.

Generative Learning and Adversarial Training

Adversarial Training

Generating Adversarial Examples:

- **Fast Gradient Sign Method (FGSM):**

Perturbs the input x in the direction of the gradient of the loss with respect to the input

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

where x' is the adversarial example, ϵ is the perturbation magnitude, J is the loss function, θ represents the model parameters, and y is the true label.

- **Projected Gradient Descent (PGD):**

An iterative method that applies multiple small perturbations while projecting the perturbed example back onto a feasible set.

$$x'_0 = x$$

$$x'_{t+1} = \text{Proj}_{\mathcal{B}(x,\epsilon)}(x'_t + \alpha \cdot \text{sign}(\nabla_x J(\theta, x'_t, y)))$$

where α is the step size, and $\mathcal{B}(x, \epsilon)$ denotes the ϵ-ball around x.

Generative Learning and Adversarial Training

Adversarial Training

Adversarial Training Process:

- **Step 1: Generate Adversarial Examples:**

During each iteration of training, generate adversarial examples from the current training data.

Generative Learning and Adversarial Training

Adversarial Training

Adversarial Training Process:

- **Step 1: Generate Adversarial Examples:**

During each iteration of training, generate adversarial examples from the current training data.

- **Step 2: Training**

Train the model on both the original and the adversarial examples.

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta\in\mathcal{S}} J(\theta, x+\delta, y) \right]$$

Where δ represents the perturbation, $\mathcal{S}$ is the set of allowed perturbations, and $\mathcal{D}$ is the data distribution.

Generative Learning and Adversarial Training

Adversarial Training

Benefits of Adversarial Training:

- **Improved Robustness:**
  - ❑ Makes models more robust to adversarial attacks.
  - ❑ Helps the model to learn to correctly classify perturbed inputs.

Generative Learning and Adversarial Training

Adversarial Training

Benefits of Adversarial Training:

- **Improved Robustness:**
  - ❑ Makes models more robust to adversarial attacks.
  - ❑ Helps the model to learn to correctly classify perturbed inputs.
- **Generalization:**
  - ❑ Improve the generalization ability of the model.
  - ❑ Learns to handle a broader range of inputs.

Generative Learning and Adversarial Training

Adversarial Training

Benefits of Adversarial Training:

- **Improved Robustness:**
  - ❑ Makes models more robust to adversarial attacks.
  - ❑ Helps the model to learn to correctly classify perturbed inputs.
- **Generalization:**
  - ❑ Improve the generalization ability of the model.
  - ❑ Learns to handle a broader range of inputs.
- **Security:**
  - ❑ Enhances the security of machine learning models (e.g., autonomous driving, medical diagnosis).

Generative Learning and Adversarial Training

Adversarial Training

Challenges of Adversarial Training:

- **Computationally Intensive:**
  - ❑ Generating adversarial examples and including them in the training is computationally expensive.

Generative Learning and Adversarial Training

Adversarial Training

Challenges of Adversarial Training:

- **Computationally Intensive:**
  - ❑ Generating adversarial examples and including them in the training is computationally expensive.
- **Trade-off with Accuracy:**
  - ❑ trade-off between robustness and accuracy on clean (non-adversarial) data.

Adversarial Training

Example :

Adversarial training algorithm using FGSM:

1. **Generate Adversarial Example:**

$$x' = x + \epsilon \cdot \mathrm{sign}(\nabla_x J(\theta, x, y))$$

2. **Training Objective:**

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[J(\theta, x, y) + J(\theta, x', y)\right]$$

In this setup, the model is trained to minimize the loss on both the original data x and the adversarial examples x'. This helps the model learn to be more resilient to adversarial perturbations, improving its robustness and security.

Generative Learning and Adversarial Training

Generative Adversarial Networks

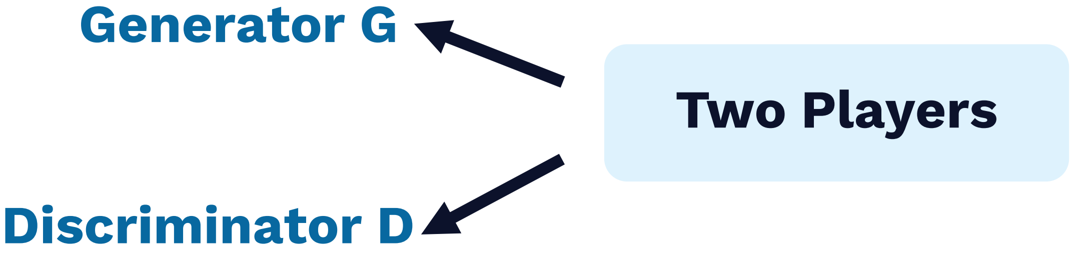
Concept:

Two Players

Generative Learning and Adversarial Training

Generative Adversarial Networks

Concept:



Generator G

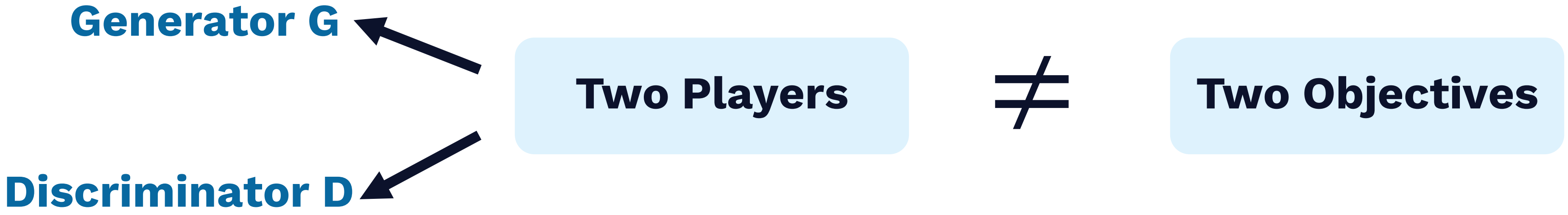
Two Players

Discriminator D

Generative Learning and Adversarial Training

Generative Adversarial Networks

Concept:



Generator G

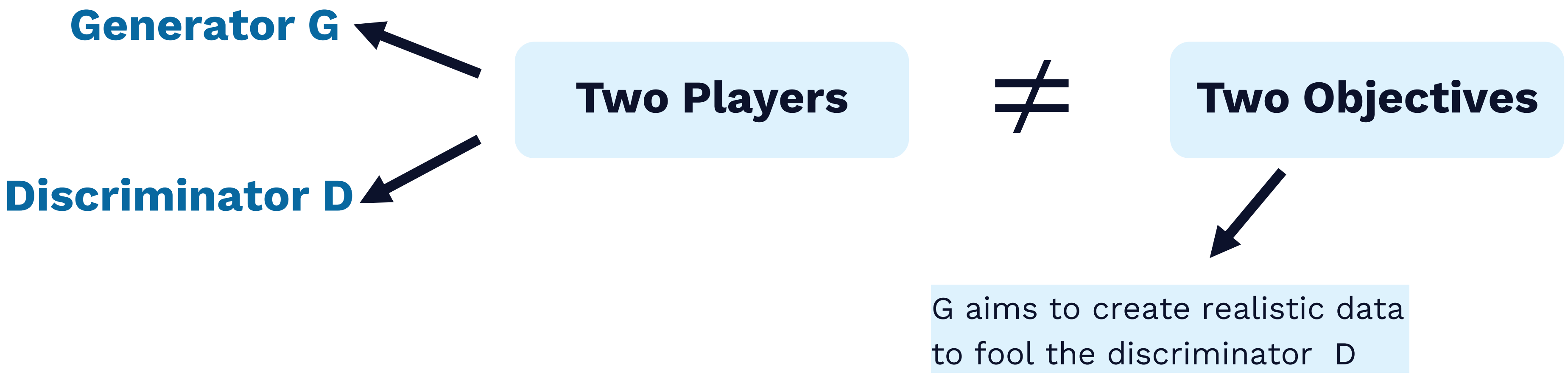
Discriminator D

Two Players ≠ **Two Objectives**

Generative Learning and Adversarial Training

Generative Adversarial Networks

Concept:



Generator G

Discriminator D

Two Players

≠

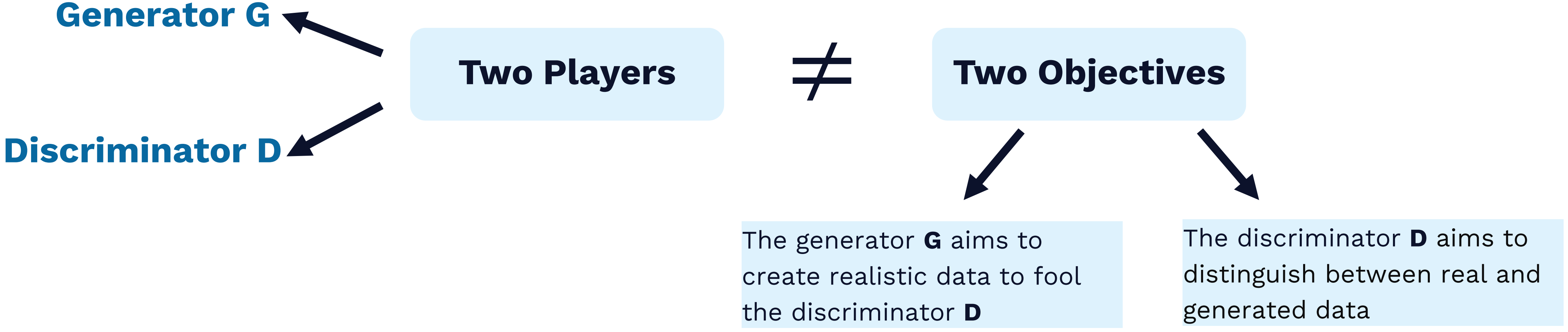
Two Objectives

G aims to create realistic data to fool the discriminator D

Generative Learning and Adversarial Training

Generative Adversarial Networks

Concept:



Generator G

Discriminator D

Two Players ≠ **Two Objectives**

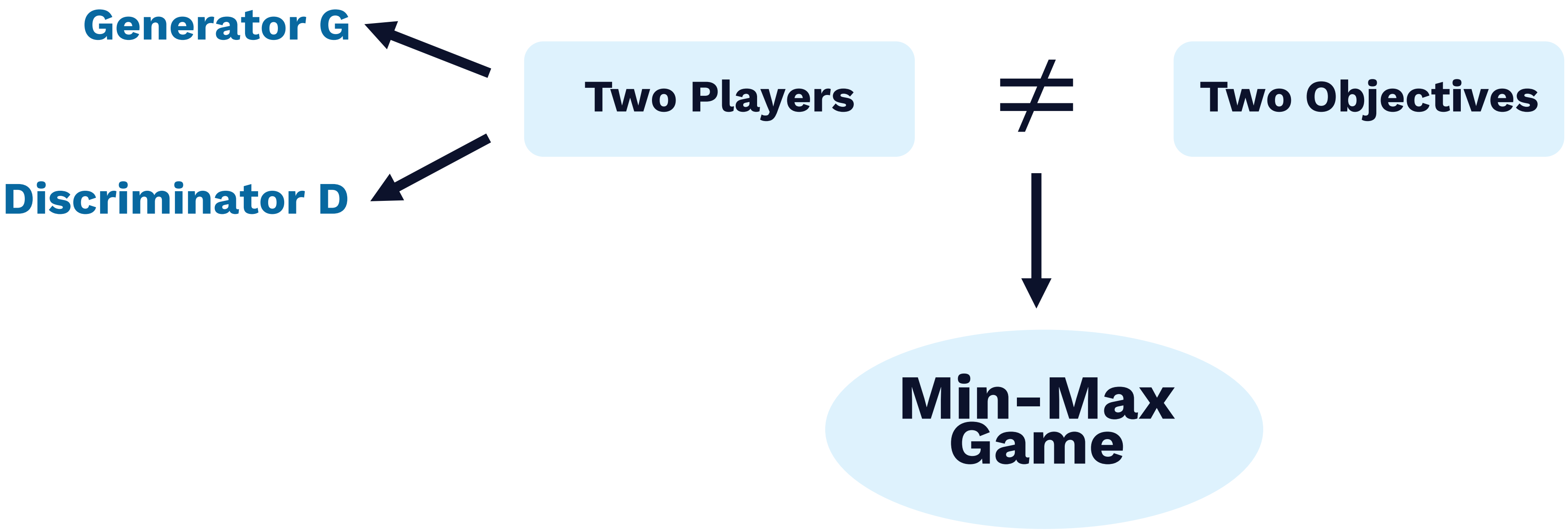
The generator **G** aims to create realistic data to fool the discriminator **D**

The discriminator **D** aims to distinguish between real and generated data

Generative Learning and Adversarial Training

Generative Adversarial Networks

Concept:



Generative Learning and Adversarial Training

Generative Adversarial Networks

Concept:

The training process of GANs can be formulated as the following **min-max** game:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

- $p_{data}(x)$ represents the distribution of the real data.
- $p_z(z)$ is the prior distribution of the noise vector z, often chosen to be a simple distribution like Gaussian or uniform.
- $G(z)$ represents the generated data from the noise vector z.
- $D(x)$ is the discriminator's estimate of the probability that x is real.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Concept:

The training process of GANs can be formulated as the following **min-max** game:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

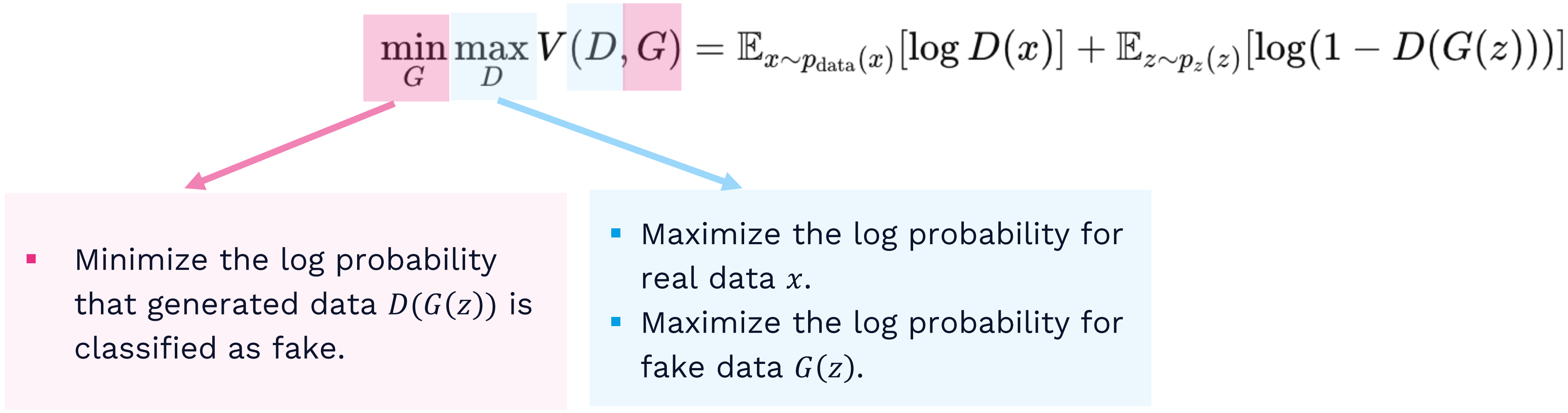
- Maximize the log probability for real data x.
- Maximize the log probability for fake data $G(z)$.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Concept:

The training process of GANs can be formulated as the following **min-max** game:



$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_{z}(z)}[\log(1 - D(G(z)))]$$

- Minimize the log probability that generated data $D(G(z))$ is classified as fake.

- Maximize the log probability for real data x.
- Maximize the log probability for fake data $G(z)$.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Training Process:

The minimax game is solved by alternating optimization steps for D and G:

Discriminator Training:

- Sample a batch of real data $\{x^{(i)}\}_{i=1}^{m}$ from the true data distribution p_{data}.
- Sample a batch of noise vectors $\{z^{(i)}\}_{i=1}^{m}$ from a prior noise distribution p_z (e.g., a Gaussian or uniform distribution).
- Generate fake data using the generator $\{G(z^{(i)}; \theta_G)\}_{i=1}^{m}$.
- Compute the discriminator loss function:

$$L_D = -\frac{1}{m} \sum_{i=1}^{m} \left[\log D(x^{(i)}; \theta_D) + \log(1 - D(G(z^{(i)}; \theta_G); \theta_D)) \right]$$

- Update the discriminator parameters θ_D using gradient descent:

$$\theta_D \leftarrow \theta_D - \eta \nabla_{\theta_D} L_D$$

Generative Adversarial Networks

Training Process:

The minimax game is solved by alternating optimization steps for D and G:

Generator Training:

- Sample a batch of noise vectors $\{z^{(i)}\}_{i=1}^{m}$ from the noise distribution p_z.
- Generate fake data $\{G(z^{(i)}; \theta_G)\}_{i=1}^{m}$.
- Compute the generator loss function:

$$L_G = -\frac{1}{m}\sum_{i=1}^{m} \log D(G(z^{(i)}; \theta_G); \theta_D)$$

- Update the generator parameters θ_G using gradient descent:

$$\theta_G \leftarrow \theta_G - \eta \nabla_{\theta_G} L_G$$

Generative Learning and Adversarial Training

Generative Adversarial Networks

Convergence:

The game reaches a Nash equilibrium when the generator produces data that is indistinguishable from real data, making the discriminator's predictions equally likely to be real or fake. At this point:

$$D(x) = \frac{1}{2} \quad \text{for all } x$$

Generative Learning and Adversarial Training

Generative Adversarial Networks

Challenges:

- **Mode Collapse:** Occurs when the generator produces a limited variety of outputs, failing to capture the diversity of the data distribution.

➡ limits the utility of GANs in applications requiring diverse outputs, as the generated samples do not represent the full range of possible data points.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Challenges:

- **Mode Collapse:** Occurs when the generator produces a limited variety of outputs, failing to capture the diversity of the data distribution.

➡ limits the utility of GANs in applications requiring diverse outputs, as the generated samples do not represent the full range of possible data points.

- **Training Instability:** The training process of GANs can be highly unstable due to the adversarial nature of the two networks. Small changes in parameters can lead to large variations in the results.

➡ Instability can cause the generator or discriminator to overpower the other, leading to poor quality generated samples and difficulty in achieving convergence.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Challenges:

- **Mode Collapse:** Occurs when the generator produces a limited variety of outputs, failing to capture the diversity of the data distribution.

  ➡ limits the utility of GANs in applications requiring diverse outputs, as the generated samples do not represent the full range of possible data points.

- **Training Instability:** The training process of GANs can be highly unstable due to the adversarial nature of the two networks. Small changes in parameters can lead to large variations in the results.

  ➡ Instability can cause the generator or discriminator to overpower the other, leading to poor quality generated samples and difficulty in achieving convergence.

- **Vanishing Gradients:** When the discriminator becomes too accurate, the gradients passed to the generator can become very small, leading to slow or stalled updates in the generator.

  ➡ This makes it challenging for the generator to improve and learn to produce better samples.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Challenges:

- **Mode Dropping:** Occurs when the generator produces a limited variety of outputs, failing to capture the diversity of the data distribution.

⇨ This results in the generated data not fully representing the variety present in the real data distribution.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Challenges:

- **Mode Dropping:** Occurs when the generator produces a limited variety of outputs, failing to capture the diversity of the data distribution.

  ➡ This results in the generated data not fully representing the variety present in the real data distribution.

- **Evaluation Metrics:** Evaluating GAN performance is difficult due to the lack of universally accepted metrics. Commonly used metrics like *Inception Score (IS)* and *Frechet Inception Distance (FID)* have limitations.

  ➡ The difficulty in evaluation makes it challenging to objectively compare different GAN models and improvements.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Challenges:

- **Mode Dropping:** Occurs when the generator produces a limited variety of outputs, failing to capture the diversity of the data distribution.

  ➡ This results in the generated data not fully representing the variety present in the real data distribution.

- **Evaluation Metrics:** Evaluating GAN performance is difficult due to the lack of universally accepted metrics. Commonly used metrics like *Inception Score (IS)* and *Frechet Inception Distance (FID)* have limitations.

  ➡ The difficulty in evaluation makes it challenging to objectively compare different GAN models and improvements.

- **Hyperparameter Sensitivity:** GANs are highly sensitive to hyperparameters such as learning rate, batch size, and network architecture.

  ➡ This requires extensive experimentation and tuning, making the training process resource-intensive and time-consuming.

Generative Adversarial Networks

Challenges:

- **Lack of Theoretical Understanding:** The theoretical underpinnings of GANs are not fully understood, particularly regarding why certain architectures or training regimes work better than others.

  ➡ This limits the ability to design more effective and stable GAN models based on theoretical insights.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Challenges:

- **Lack of Theoretical Understanding:** The theoretical underpinnings of GANs are not fully understood, particularly regarding why certain architectures or training regimes work better than others.

  ➡ This limits the ability to design more effective and stable GAN models based on theoretical insights.

- **Computational Cost:** Training GANs, especially with large models and datasets, requires substantial computational resources.

  ➡ This limits accessibility for researchers and practitioners with limited resources and can slow down the experimentation process.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Challenges:

- **Lack of Theoretical Understanding:** The theoretical underpinnings of GANs are not fully understood, particularly regarding why certain architectures or training regimes work better than others.

  ⇨ This limits the ability to design more effective and stable GAN models based on theoretical insights.

- **Computational Cost:** Training GANs, especially with large models and datasets, requires substantial computational resources.

  ⇨ This limits accessibility for researchers and practitioners with limited resources and can slow down the experimentation process.

- **Bias and Fairness :** GANs can inadvertently learn and amplify biases present in the training data.

  ⇨ This can lead to biased and unfair generated samples, posing ethical concerns and limiting the applicability of GANs in sensitive domains.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Challenges:

- **Adversarial Attacks:** GANs are vulnerable to adversarial attacks where small perturbations in the input can lead to significant changes in the output.

  ➡ This can compromise the robustness and reliability of GAN-generated data in practical applications.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Popular GAN-based models and architectures

- **DCGAN (Deep Convolutional GAN):**
  - **Architecture:** Uses convolutional layers in the generator and discriminator.
  - **Contributions:** Introduced stable architectures for GANs and demonstrated the ability to generate realistic images from random noise.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv preprint arXiv:1511.06434.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Popular GAN-based models and architectures

- **DCGAN (Deep Convolutional GAN):**
  - **Architecture:** Uses convolutional layers in the generator and discriminator.
  - **Contributions:** Introduced stable architectures for GANs and demonstrated the ability to generate realistic images from random noise.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv preprint arXiv:1511.06434.

- **CGAN (Conditional GAN):**
  - **Architecture:** Adds extra information (e.g., class labels) to both the generator and discriminator.
  - **Contributions:** Allows control over the output generation process, making it possible to generate specific types of images.

Mirza, M., & Osindero, S. (2014). Conditional Generative Adversarial Nets. arXiv preprint arXiv:1411.1784.

Generative Adversarial Networks

Popular GAN-based models and architectures

- **DCGAN (Deep Convolutional GAN):**
  - **Architecture:** Uses convolutional layers in the generator and discriminator.
  - **Contributions:** Introduced stable architectures for GANs and demonstrated the ability to generate realistic images from random noise.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv preprint arXiv:1511.06434.

- **CGAN (Conditional GAN):**
  - **Architecture:** Adds extra information (e.g., class labels) to both the generator and discriminator.
  - **Contributions:** Allows control over the output generation process, making it possible to generate specific types of images.

Mirza, M., & Osindero, S. (2014). Conditional Generative Adversarial Nets. arXiv preprint arXiv:1411.1784.

- **WGAN (Wasserstein GAN):**
  - **Architecture:** Introduces a new loss function based on the Wasserstein distance.
  - **Contributions:** Improves training stability and provides a meaningful loss metric for GANs.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. arXiv preprint arXiv:1701.07875.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Popular GAN-based models and architectures

➢ **WGAN-GP (Wasserstein GAN with Gradient Penalty):**

- **Architecture:** An improvement over WGAN, WGAN-GP introduces a gradient penalty term to enforce the Lipschitz constraint.

Contributions: Stabilizes GAN training and improves the quality of generated samples.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved Training of Wasserstein GANs. arXiv preprint arXiv:1704.00028.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Popular GAN-based models and architectures

- **WGAN-GP (Wasserstein GAN with Gradient Penalty):**
  - **Architecture:** An improvement over WGAN, WGAN-GP introduces a gradient penalty term to enforce the Lipschitz constraint.

  Contributions: Stabilizes GAN training and improves the quality of generated samples.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved Training of Wasserstein GANs. arXiv preprint arXiv:1704.00028.

- **Pix2Pix:**
  - **Architecture:** Uses conditional GANs for paired image-to-image translation.
  - **Contributions:** Demonstrates high-quality image transformation tasks such as converting sketches to photos.

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. arXiv preprint arXiv:1611.07004.

Generative Learning and Adversarial Training

Generative Adversarial Networks

Popular GAN-based models and architectures

- **WGAN-GP (Wasserstein GAN with Gradient Penalty):**
  - **Architecture:** An improvement over WGAN, WGAN-GP introduces a gradient penalty term to enforce the Lipschitz constraint.

  Contributions: Stabilizes GAN training and improves the quality of generated samples.

  Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved Training of Wasserstein GANs. arXiv preprint arXiv:1704.00028.

- **Pix2Pix:**
  - **Architecture:** Uses conditional GANs for paired image-to-image translation.
  - **Contributions:** Demonstrates high-quality image transformation tasks such as converting sketches to photos.

  Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. arXiv preprint arXiv:1611.07004.

- **BigGAN**
  - **Architecture:** Scales up the GAN model architecture to improve the quality and diversity of generated images.
  - **Contributions:** Demonstrates the ability to generate images of unprecedented quality and diversity.

  Brock, A., Donahue, J., & Simonyan, K. (2018). Large Scale GAN Training for High Fidelity Natural Image Synthesis. arXiv preprint arXiv:1809.11096

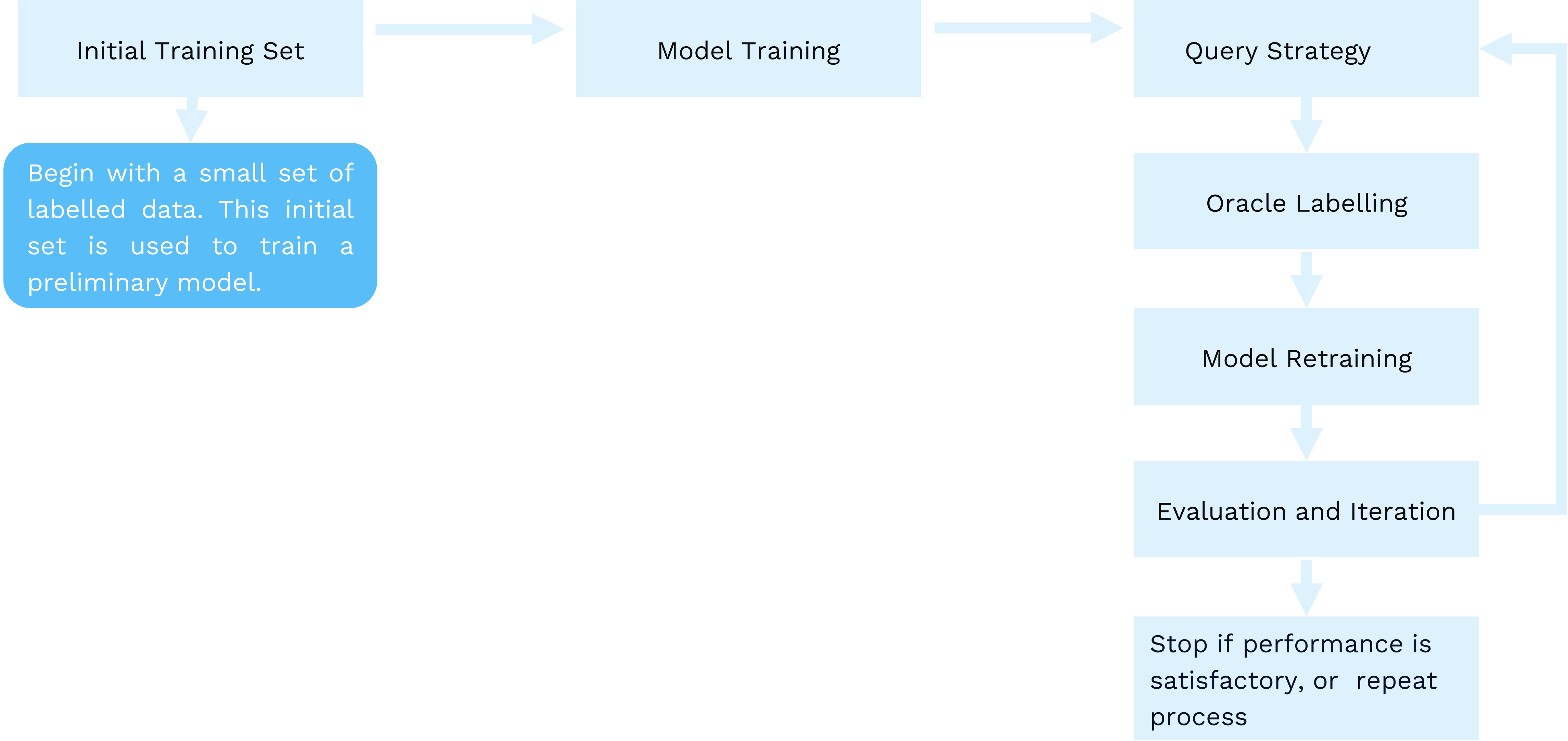
Active Learning

General Definition:

Active learning is an iterative process designed to improve a machine learning model by strategically selecting the most informative data points to be labeled by an oracle (e.g., a human annotator). The goal is to achieve high model performance with fewer labeled examples than traditional learning methods.

Active Learning

Process:



Initial Training Set

Model Training

Query Strategy

Begin with a small set of labelled data. This initial set is used to train a preliminary model.

Oracle Labelling

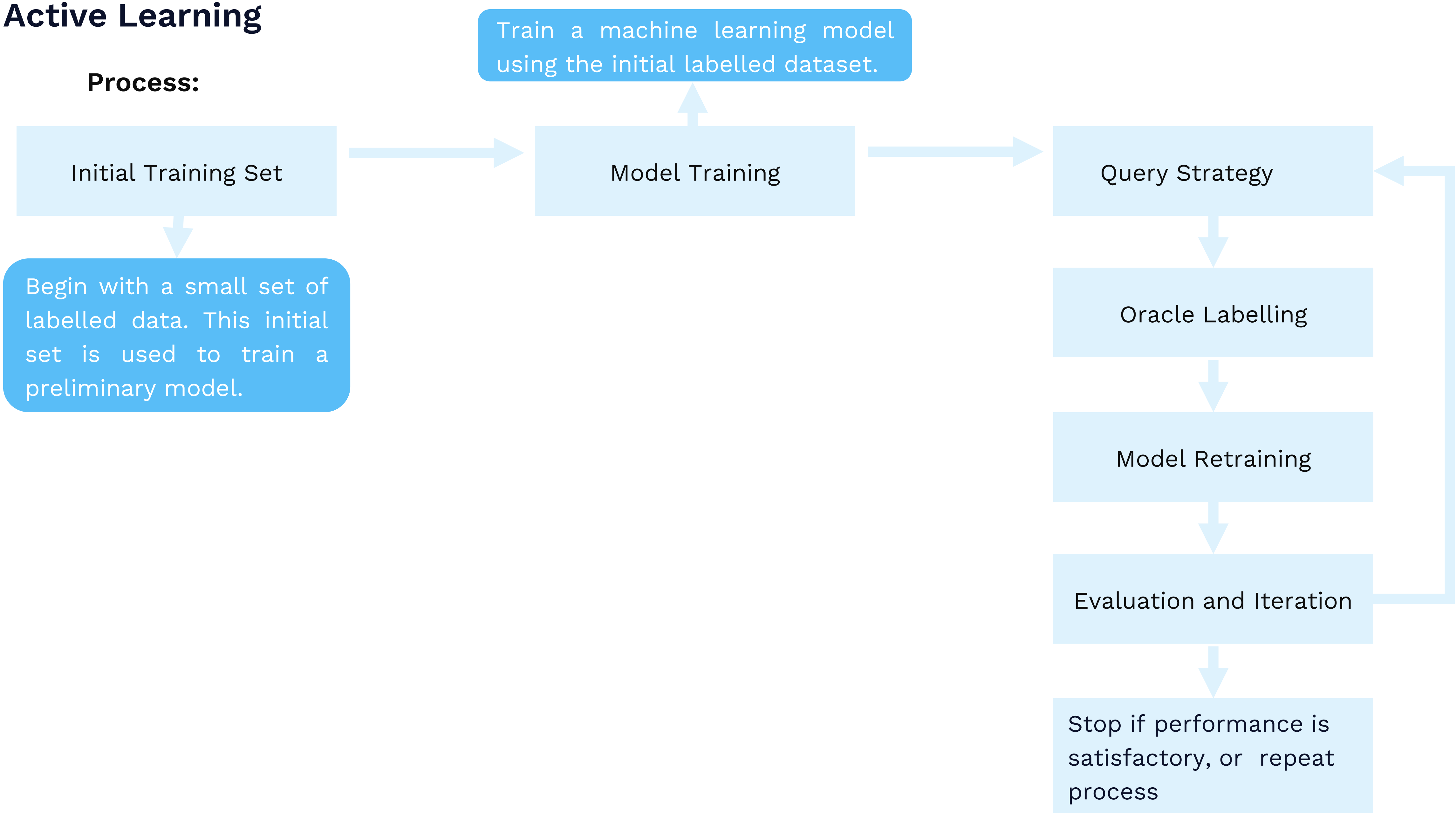
Model Retraining

Evaluation and Iteration

Stop if performance is satisfactory, or repeat process

Active Learning

Process:



Initial Training Set

Begin with a small set of labelled data. This initial set is used to train a preliminary model.

Model Training

Train a machine learning model using the initial labelled dataset.

Query Strategy

Oracle Labelling

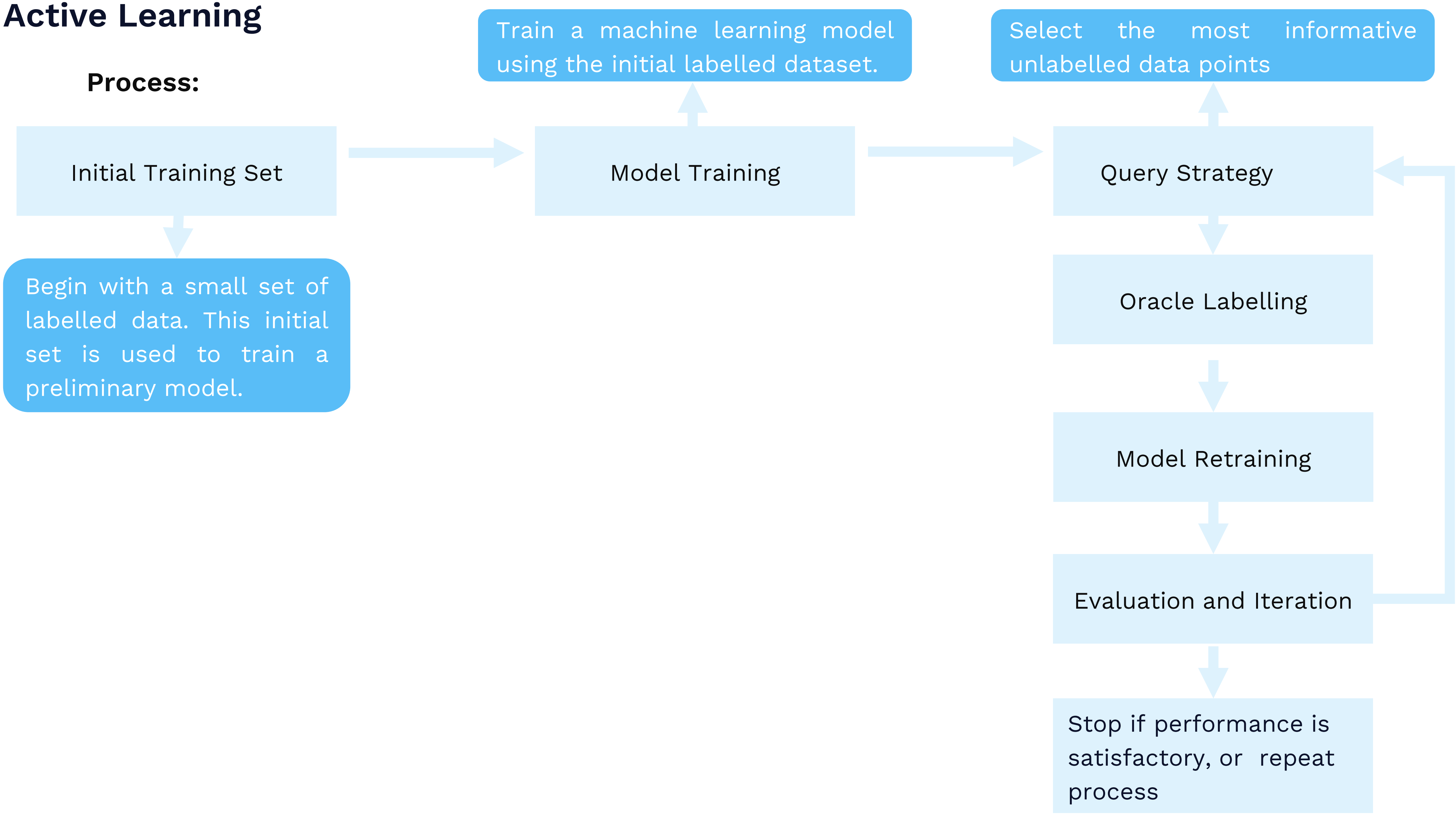
Model Retraining

Evaluation and Iteration

Stop if performance is satisfactory, or repeat process

Active Learning

Process:



Initial Training Set

Begin with a small set of labelled data. This initial set is used to train a preliminary model.

Model Training

Train a machine learning model using the initial labelled dataset.

Query Strategy

Select the most informative unlabelled data points

Oracle Labelling

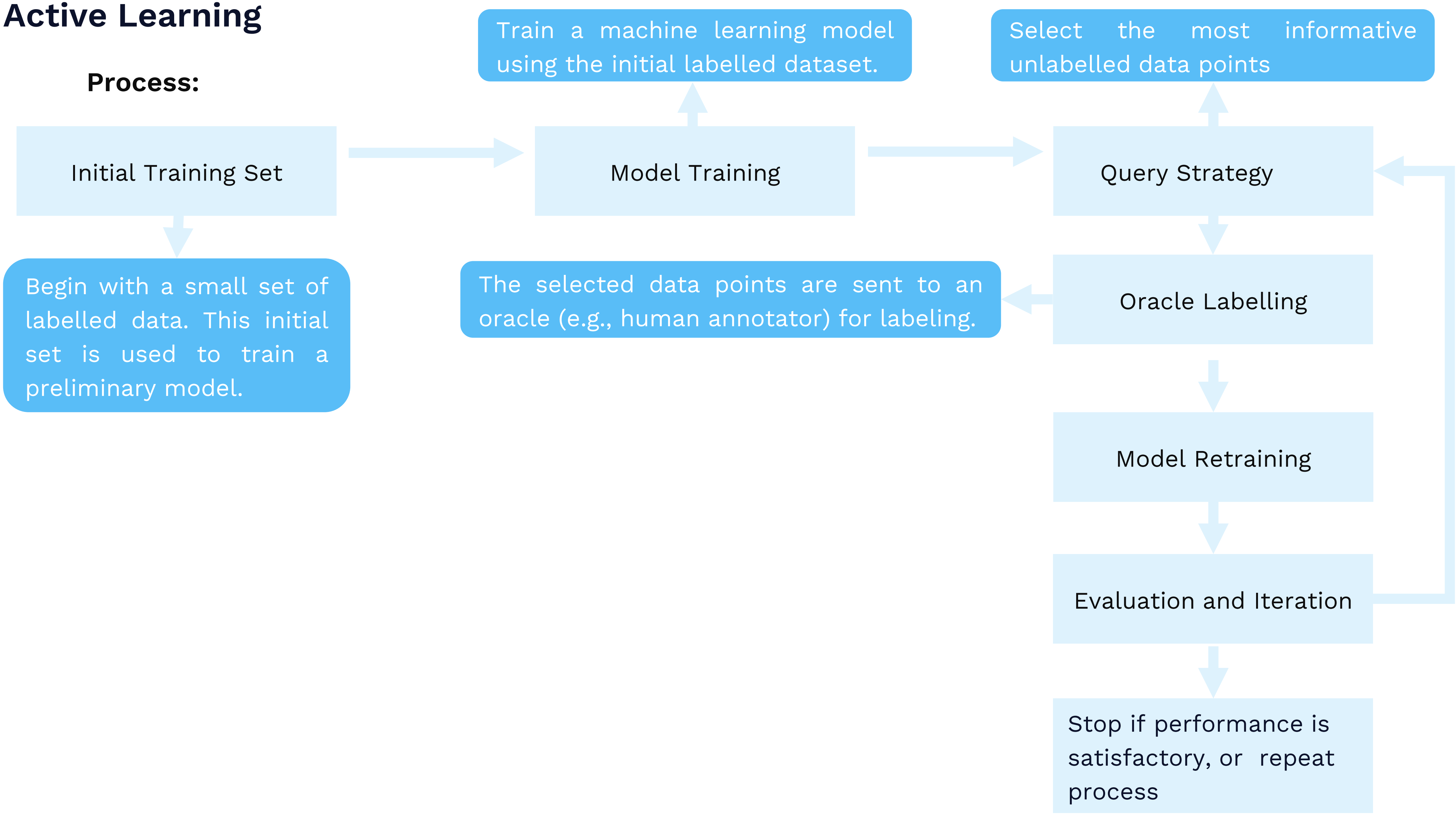
Model Retraining

Evaluation and Iteration

Stop if performance is satisfactory, or repeat process

Active Learning

Process:



Initial Training Set

Begin with a small set of labelled data. This initial set is used to train a preliminary model.

Model Training

Train a machine learning model using the initial labelled dataset.

Query Strategy

Select the most informative unlabelled data points

Oracle Labelling

The selected data points are sent to an oracle (e.g., human annotator) for labeling.

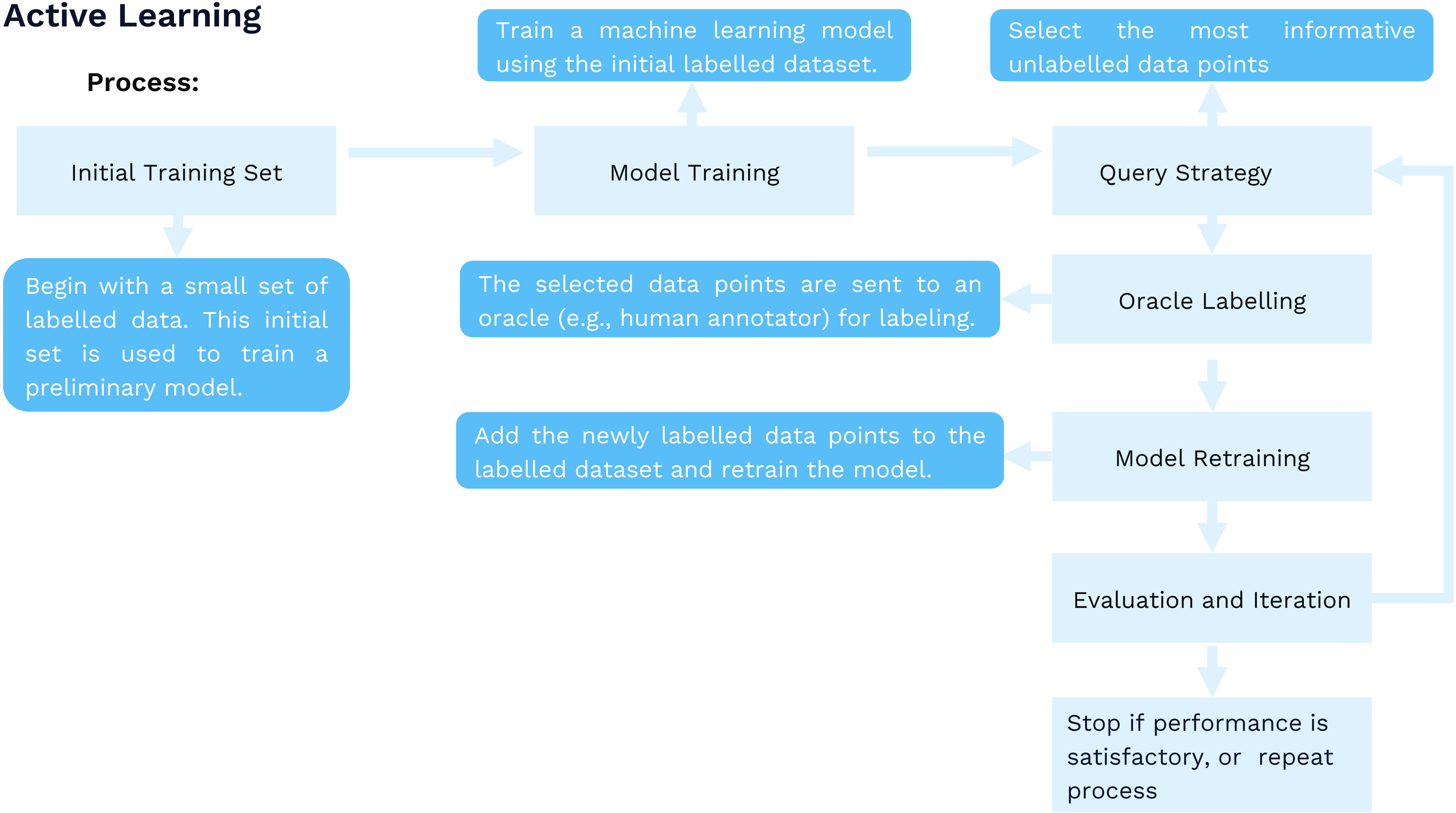
Model Retraining

Evaluation and Iteration

Stop if performance is satisfactory, or repeat process

Active Learning

Process:



Initial Training Set

Begin with a small set of labelled data. This initial set is used to train a preliminary model.

Model Training

Train a machine learning model using the initial labelled dataset.

Query Strategy

Select the most informative unlabelled data points

Oracle Labelling

The selected data points are sent to an oracle (e.g., human annotator) for labeling.

Model Retraining

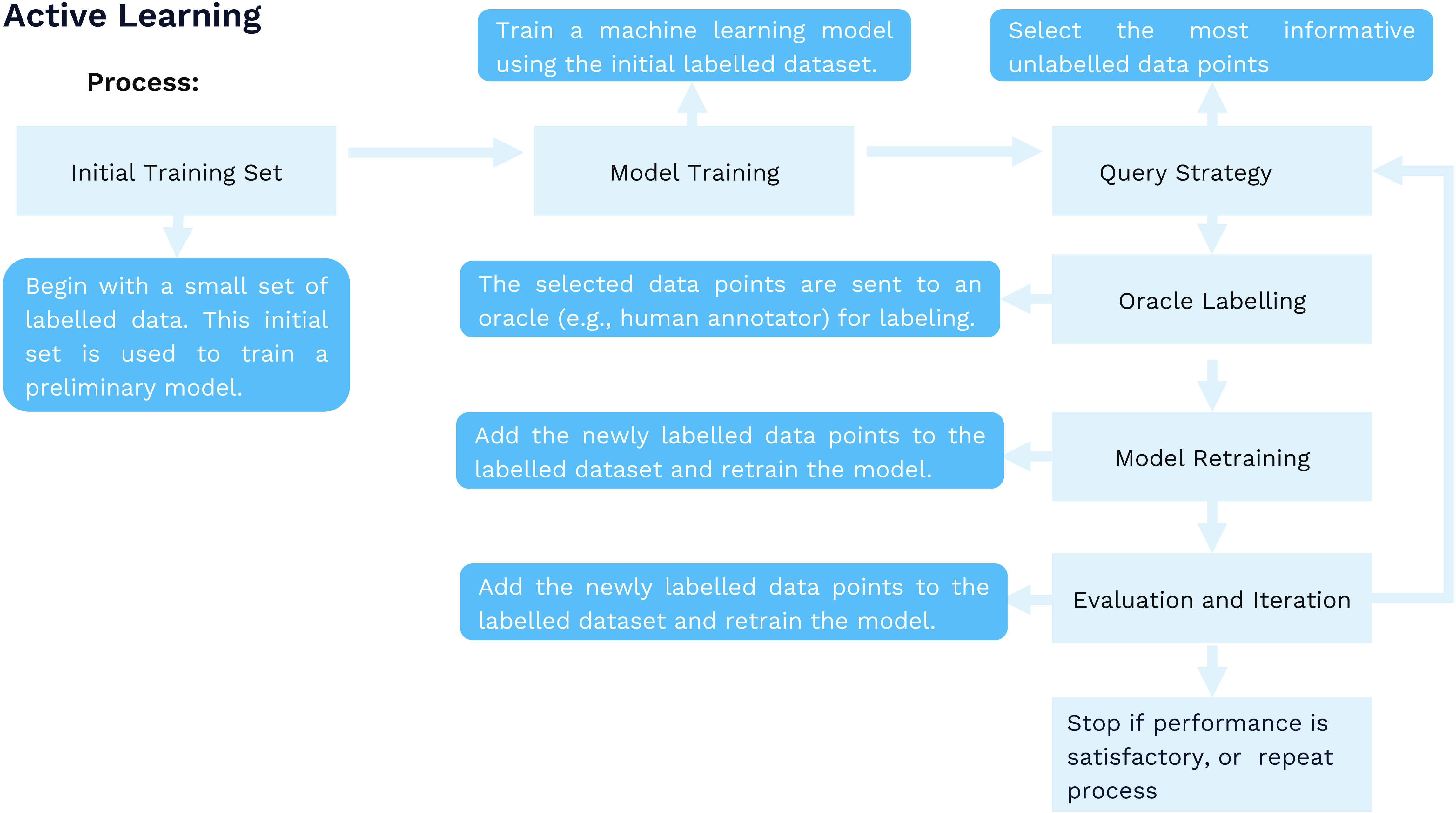
Add the newly labelled data points to the labelled dataset and retrain the model.

Evaluation and Iteration

Stop if performance is satisfactory, or repeat process

Active Learning

Process:



Initial Training Set

Begin with a small set of labelled data. This initial set is used to train a preliminary model.

Model Training

Train a machine learning model using the initial labelled dataset.

Query Strategy

Select the most informative unlabelled data points

Oracle Labelling

The selected data points are sent to an oracle (e.g., human annotator) for labeling.

Model Retraining

Add the newly labelled data points to the labelled dataset and retrain the model.

Evaluation and Iteration

Add the newly labelled data points to the labelled dataset and retrain the model.

Stop if performance is satisfactory, or repeat process

Active Learning

Popular Query Strategies:

- **Uncertainty Sampling:**

Description:

- Selects samples for which it is least confident about the output. Various metrics are used to measure uncertainty, such as:

  - ✓ **Margin Sampling**: Chooses the sample where the difference between the first and second most probable classes is smallest.
  - ✓ **Entropy:** Measures the uncertainty in the probability distribution output by the model.
  - ✓ **Least Confident Sampling:** Selects the sample with the lowest predicted probability for the most likely class.

Active Learning

Popular Query Strategies:

➢ Uncertainty Sampling:

Description:

- Selects samples for which it is least confident about the output. Various metrics are used to measure uncertainty, such as:

  - ✓ **Margin Sampling**: Chooses the sample where the difference between the first and second most probable classes is smallest.
  - ✓ **Entropy:** Measures the uncertainty in the probability distribution output by the model.
  - ✓ **Least Confident Sampling:** Selects the sample with the lowest predicted probability for the most likely class.

Limitations:

- Can focus too much on outliers or noisy data.
- Can overlook representative samples.

Active Learning

Popular Query Strategies:

➢ Query by Committee (QBC) (1/2):

Description:

- Uses an ensemble of models (the committee) trained on the current labeled dataset. The samples about which the committee members disagree the most are selected for labeling.

✓ **Vote Entropy**: Measures the entropy of the votes cast by each committee member for a particular sample. Higher entropy indicates more disagreement.

$$\text{Vote Entropy}(x) = -\sum_{c \in \mathcal{C}} \frac{v_c(x)}{N} \log \left(\frac{v_c(x)}{N} \right)$$

Where $v_c(x)$ is the number of votes for class c for sample x, and N is the total number of committee members.

✓ **Kullback-Leibler (KL) Divergence:** Measures the divergence between the probability distributions predicted by the committee members for a particular sample.

$$\text{KL-Divergence}(P \parallel Q) = \sum_{i} P(i) \log \left(\frac{P(i)}{Q(i)} \right)$$

Where P and Q are the probability distributions predicted by two different committee members.

Active Learning

Popular Query Strategies:

➢ Query by Committee (QBC) (2/2):

Description:

- ✓ **Disagreement Ratio:** Measures the proportion of committee members that disagree with the majority vote for a particular sample.

$$\text{Disagreement Ratio}(x) = 1 - \frac{\max_{c \in \mathcal{C}} v_c(x)}{N}$$

Where $v_c(x)$ is the number of votes for class c for sample x, and N is the total number of committee members.

- ✓ **Variance:** Measures the variance of the predicted probabilities for a particular sample across the committee members.

$$\text{Variance}(x) = \frac{1}{N}\sum_{i=1}^{N}(p_i(x) - \bar{p}(x))^2$$

Where $p_i(\mathrm{x})$ is the probability predicted by the $i-th$ committee member for sample x, and $\bar{p}(\mathrm{x})$ is the average predicted probability for sample x.

Active Learning

Popular Query Strategies:

Query by Committee (QBC) (2/2):

Description:

- ✓ **Disagreement Ratio:** Measures the proportion of committee members that disagree with the majority vote for a particular sample.

$$\text{Disagreement Ratio}(x) = 1 - \frac{\max_{c \in \mathcal{C}} v_c(x)}{N}$$

Where $v_c(x)$ is the number of votes for class c for sample x, and N is the total number of committee members.

- ✓ **Variance:** Measures the variance of the predicted probabilities for a particular sample across the committee members.

$$\text{Variance}(x) = \frac{1}{N} \sum_{i=1}^{N} (p_i(x) - \bar{p}(x))^2$$

Where $p_i(\mathrm{x})$ is the probability predicted by the $i-th$ committee member for sample x, and $\bar{p}(\mathrm{x})$ is the average predicted probability for sample x.

Limitations:

- Computationally expensive due to maintaining multiple models.
- Requires a diverse committee to be effective, which can be challenging to achieve.

Active Learning

Popular Query Strategies:

- **Expected Model Change:**

Description: Selects samples that would result in the greatest change to the current model if labeled and added to the training set.

1. Compute the Probability Distribution: Use the current model to compute $p(y|x,\theta)$, the probability distribution over possible labels for each candidate sample x.
2. Estimate Parameter Updates: For each possible label y, estimate the updated parameters θ' by performing a hypothetical training step using the sample (x,y).
3. Calculate the Change: Measure the change in the parameters $\|\theta'-\theta\|$.
4. Compute the Expectation: Average the measured changes weighted by their probabilities $p(y|x,\theta)$.

Active Learning

Popular Query Strategies:

- **Expected Model Change:**

Description: Selects samples that would result in the greatest change to the current model if labeled and added to the training set.

1. Compute the Probability Distribution: Use the current model to compute p(y|x,θ), the probability distribution over possible labels for each candidate sample x.
2. Estimate Parameter Updates: For each possible label y, estimate the updated parameters θ′ by performing a hypothetical training step using the sample (x,y).
3. Calculate the Change: Measure the change in the parameters ‖θ′−θ‖.
4. Compute the Expectation: Average the measured changes weighted by their probabilities p(y|x,θ).

Limitations:

- Computationally intensive as it requires estimating the impact of each candidate sample on the model.
- Assumes that the model change will always lead to performance improvement, which may not always be the case.

Active Learning

Popular Query Strategies:

- **Expected Error Reduction:**

Description: Chooses samples that are expected to most reduce the overall prediction error of the model.

Active Learning

Popular Query Strategies:

- **Expected Error Reduction:**

Description: Chooses samples that are expected to most reduce the overall prediction error of the model.

Limitations:

- Requires estimating the error reduction for each candidate, which is computationally expensive.
- The estimation process itself might be prone to errors, affecting the sample selection quality.

Active Learning

Popular Query Strategies:

- **Expected Error Reduction:**

Description: Chooses samples that are expected to most reduce the overall prediction error of the model.

Limitations:

- Requires estimating the error reduction for each candidate, which is computationally expensive.
- The estimation process itself might be prone to errors, affecting the sample selection quality.

- **Diversity Sampling:**

Description: Selects samples that are not only uncertain but also diverse, ensuring that the labeled dataset covers different regions of the data space.

Active Learning

Popular Query Strategies:

➢ Expected Error Reduction:

Description: Chooses samples that are expected to most reduce the overall prediction error of the model.

Limitations:

- Requires estimating the error reduction for each candidate, which is computationally expensive.
- The estimation process itself might be prone to errors, affecting the sample selection quality.

➢ Diversity Sampling:

Description: Selects samples that are not only uncertain but also diverse, ensuring that the labeled dataset covers different regions of the data space.

Limitations:

- Balancing between uncertainty and diversity can be challenging.
- Computationally intensive due to the need for measuring diversity.

Active Learning

Popular Query Strategies:

- **Density-Weighted Methods:**

Description: Selects samples based on a combination of uncertainty and their representativeness within the data distribution, often measured using density estimates.

Active Learning

Popular Query Strategies:

- **Density-Weighted Methods:**

Description: Selects samples based on a combination of uncertainty and their representativeness within the data distribution, often measured using density estimates.

Limitations:

- Requires Calculating density estimates can be computationally expensive.
- Density estimation might not always be accurate, leading to suboptimal sample selection.

Active Learning

Limitations of Active Learning:

- **Computational Cost**:

  Many active learning methods require extensive computations, such as multiple model training (in QBC) or error estimations (in Expected Error Reduction), which can be resource-intensive and time-consuming.

Active Learning

Limitations of Active Learning:

- **Computational Cost**:

  Many active learning methods require extensive computations, such as multiple model training (in QBC) or error estimations (in Expected Error Reduction), which can be resource-intensive and time-consuming.

- **Scalability**:

  Active learning methods might struggle to scale with large datasets or high-dimensional data due to the computational demands.

Active Learning

Limitations of Active Learning:

- **Computational Cost**:

  Many active learning methods require extensive computations, such as multiple model training (in QBC) or error estimations (in Expected Error Reduction), which can be resource-intensive and time-consuming.

- **Scalability**:

  Active learning methods might struggle to scale with large datasets or high-dimensional data due to the computational demands.

- **Dependency on Initial Labelled Data**:

  The performance of active learning can be heavily influenced by the initial set of labelled data. Poor initial samples can lead to suboptimal model performance and poor sample selection.

Active Learning

Limitations of Active Learning:

- **Computational Cost**:

  Many active learning methods require extensive computations, such as multiple model training (in QBC) or error estimations (in Expected Error Reduction), which can be resource-intensive and time-consuming.

- **Scalability**:

  Active learning methods might struggle to scale with large datasets or high-dimensional data due to the computational demands.

- **Dependency on Initial Labelled Data**:

  The performance of active learning can be heavily influenced by the initial set of labelled data. Poor initial samples can lead to suboptimal model performance and poor sample selection.

- **Noisy Data Sensitivity**:

  Active learning can sometimes focus too much on uncertain or noisy data, leading to poor generalization if not properly managed.

Active Learning

Limitations of Active Learning:

- **Diversity-Accuracy Trade-off**:

  Balancing the trade-off between selecting highly uncertain samples and maintaining diversity in the dataset is challenging and crucial for effective learning.

Active Learning

Limitations of Active Learning:

- **Diversity-Accuracy Trade-off**:

  Balancing the trade-off between selecting highly uncertain samples and maintaining diversity in the dataset is challenging and crucial for effective learning.

- **Annotation Effort**:

  While active learning aims to minimize labelling effort, the annotation process can still be time-consuming, especially for complex tasks requiring expert knowledge.

Active Learning

Limitations of Active Learning:

- **Diversity-Accuracy Trade-off**:

  Balancing the trade-off between selecting highly uncertain samples and maintaining diversity in the dataset is challenging and crucial for effective learning.

- **Annotation Effort**:

  While active learning aims to minimize labelling effort, the annotation process can still be time-consuming, especially for complex tasks requiring expert knowledge.

- **Model Dependency**:

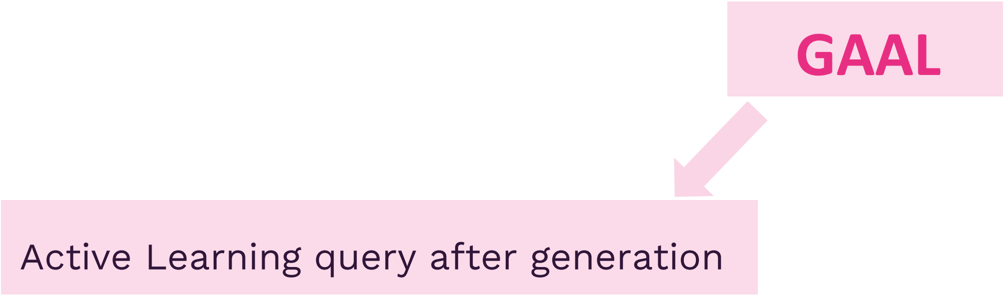
  The effectiveness of an active learning strategy can be highly dependent on the underlying model. Some models may not show significant performance gains with active learning compared to random sampling.

Generative Adversarial Active Learning (GAAL)

- The main idea is to use active learning to guide the generator to generate synthetic examples that are the most informative.

Generative Adversarial Active Learning (GAAL)

- The main idea is to use active learning to guide the generator to generate synthetic examples that are the most informative.

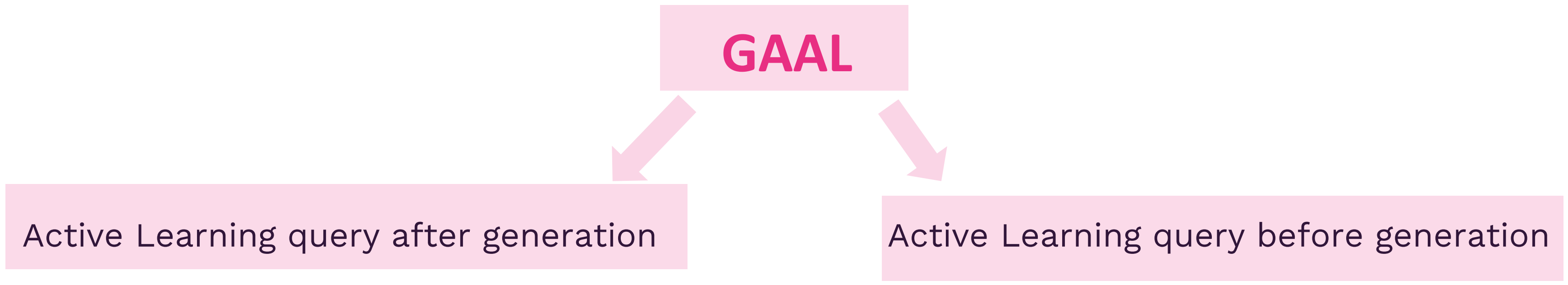


GAAL

Active Learning query after generation

Generative Adversarial Active Learning (GAAL)

- The main idea is to use active learning to guide the generator to generate synthetic examples that are the most informative.



GAAL

Active Learning query after generation

Active Learning query before generation

Generative Adversarial Active Learning (GAAL)

Active Learning query after generation

1. Generate Synthetic Data using the adversarial training procedure

$$x' = G(z), \quad z \sim p_z(z)$$

Generative Adversarial Active Learning (GAAL)

Active Learning query after generation

1. Generate Synthetic Data using the adversarial training procedure

$$x' = G(z), \quad z \sim p_z(z)$$

2. Compute Informativeness Score

$$S(x) = \alpha \cdot \mathrm{Uncertainty}(x) + \beta \cdot \mathrm{Diversity}(x)$$

Where α and β are weights, and $S(x)$ combines uncertainty and diversity measures.

Generative Adversarial Active Learning (GAAL)

Active Learning query after generation

1. Generate Synthetic Data using the adversarial training procedure

$$x' = G(z), \quad z \sim p_z(z)$$

2. Compute Informativeness Score

$$S(x) = \alpha \cdot \text{Uncertainty}(x) + \beta \cdot \text{Diversity}(x)$$

Where α and β are weights, and $S(x)$ combines uncertainty and diversity measures.

3. Select Samples for Labeling:

$$x^* = \arg \max_{x \in \mathcal{U} \cup G(\mathcal{Z})} S(x)$$

where $\mathcal{U}$ is the pool of unlabelled real samples and $G(\mathcal{Z})$ is the set of synthetic samples generated by the GAN.

Generative Adversarial Active Learning (GAAL)

Active Learning query before generation

Main Idea: Guiding the generator produce samples that are likely to be **the most informative.**

Generative Adversarial Active Learning (GAAL)

Active Learning query before generation

Main Idea: Guiding the generator produce samples that are likely to be **the most informative.**

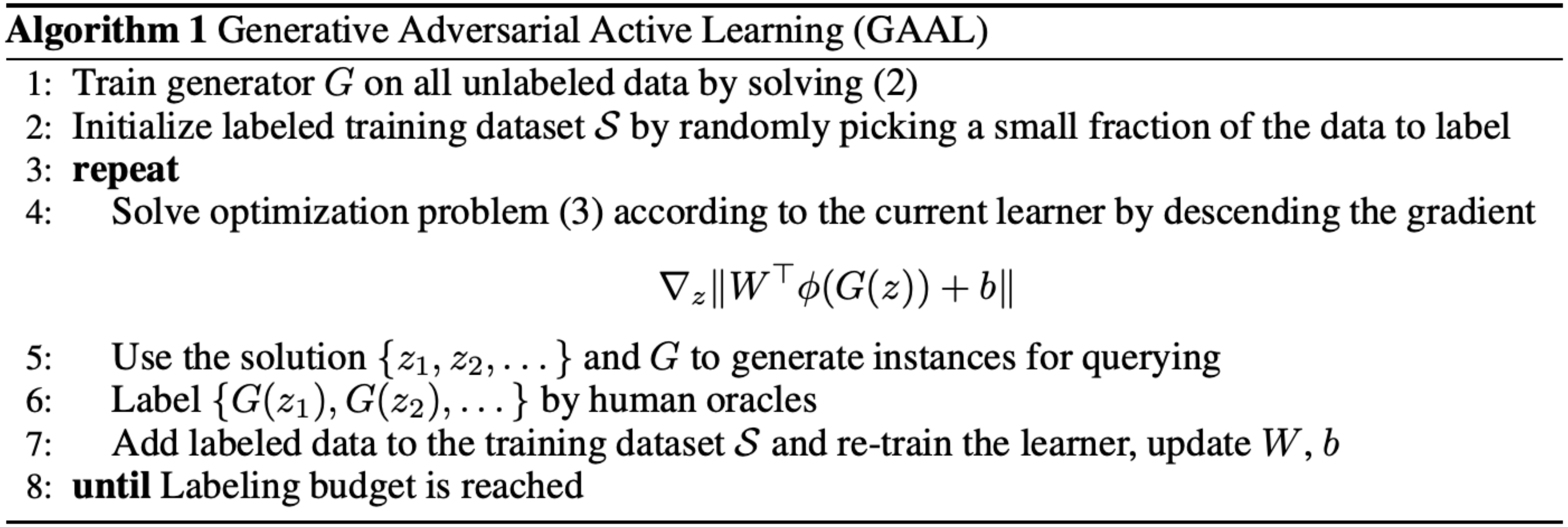
Example: Generate training examples that are likely to be on the decision boundary for an SVM classifier.

Generative Adversarial Active Learning (GAAL)

Active Learning query before generation

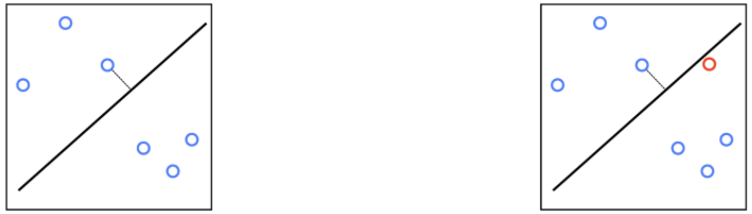
Main Idea: Guiding the generator produce samples that are likely to be **the most informative.**

Example: Generate training examples that are likely to be on the decision boundary for an SVM classifier.



Algorithm 1 Generative Adversarial Active Learning (GAAL)

```
Train generator $G$ on all unlabeled data by solving (2)
Initialize labeled training dataset $\mathcal{S}$ by randomly picking a small fraction of the data to label
repeat
    Solve optimization problem (3) according to the current learner by descending the gradient
        $\nabla_z \|W^\top \phi(G(z)) + b\|$
    Use the solution $\{z_1, z_2, \dots\}$ and $G$ to generate instances for querying
    Label $\{G(z_1), G(z_2), \dots\}$ by human oracles
    Add labeled data to the training dataset $\mathcal{S}$ and re-train the learner, update $W, b$
until Labeling budget is reached
```



(a) SVM$_{active}$

(b) GAAL

Zhu, Jia-Jie, and José Bento. "Generative adversarial active learning." arXiv preprint arXiv:1702.07956 (2017).

GAAL Impact:

Advantages

- **Sample Efficiency:**

By generating and selecting the most informative samples, GAAL reduces the amount of labeled and generated data needed to achieve high performance.

GAAL Impact:

Advantages

- **Sample Efficiency:**

By generating and selecting the most informative samples, GAAL reduces the amount of labeled and generated data needed to achieve high performance.

- **Improved Learning:**

The generated samples can cover regions of the data space that are underrepresented, leading to a more robust model.

GAAL Impact:

Advantages

- **Sample Efficiency:**

By generating and selecting the most informative samples, GAAL reduces the amount of labeled and generated data needed to achieve high performance.

- **Improved Learning:**

The generated samples can cover regions of the data space that are underrepresented, leading to a more robust model.

- **Dynamic Adaptation:**

GAAL can adapt to changes in the data distribution over time, improving its applicability in real-world scenarios.

GAAL Impact:

Limitations

- **Training Complexity:**

Training GANs is computationally intensive and can be unstable. The integration with active learning adds further complexity.

GAAL Impact:

Limitations

- **Training Complexity:**

Training GANs is computationally intensive and can be unstable. The integration with active learning adds further complexity.

- **Quality of Generated Samples:**

The effectiveness of GAAL heavily depends on the quality of the samples generated by �G. Poor quality samples can negatively impact the learning process.

Concluding Remarks:

Active Learning and GANs Synergy

- ✓ Active learning and Generative Adversarial Networks (GANs) form a powerful combination, enabling efficient model training with fewer labeled examples.
- ✓ Active learning prioritizes the most informative samples, enhancing the effectiveness of GANs in various tasks.

Concluding Remarks:

Important Considerations for Future Work:

1. Resource Consumption:

- Future research should prioritize optimizing resource consumption to make active learning and GANs more scalable and accessible.
- Efficient algorithms and models can significantly reduce computational costs and energy consumption.

Concluding Remarks:

Important Considerations for Future Work:

1. **Resource Consumption:**

   - Future research should prioritize optimizing resource consumption to make active learning and GANs more scalable and accessible.

   - Efficient algorithms and models can significantly reduce computational costs and energy consumption.

2. **Trustworthiness and Robustness:**

   - Ensuring the trustworthiness of models is crucial, especially in sensitive applications like healthcare and finance.

   - Robust models that can withstand adversarial attacks and input variations are essential for reliable performance.

Concluding Remarks:

Important Considerations for Future Work:

1. **Resource Consumption:**
   - ➢ Future research should prioritize optimizing resource consumption to make active learning and GANs more scalable and accessible.
   - ➢ Efficient algorithms and models can significantly reduce computational costs and energy consumption.
2. **Trustworthiness and Robustness:**
   - ➢ Ensuring the trustworthiness of models is crucial, especially in sensitive applications like healthcare and finance.
   - ➢ Robust models that can withstand adversarial attacks and input variations are essential for reliable performance.
3. **Explainability:**
   - ➢ Explainability is a key factor in gaining user trust and understanding model decisions.
   - ➢ Developing methods to interpret and explain the decisions of active learning models and GANs will enhance their adoption and transparency.

LAMARR

INSTITUTE FOR
MACHINE LEARNING
AND ARTIFICIAL
INTELLIGENCE

Generative Adversarial Networks and Active Learning

Dr. Amal Saadallah

Partner institutions:

technische universität dortmund

Fraunhofer IAIS

UNIVERSITÄT BONN

Fraunhofer IML

Institutionally funded by:

Bundesministerium
für Bildung
und Forschung

Ministerium für
Kultur und Wissenschaft
des Landes Nordrhein-Westfalen