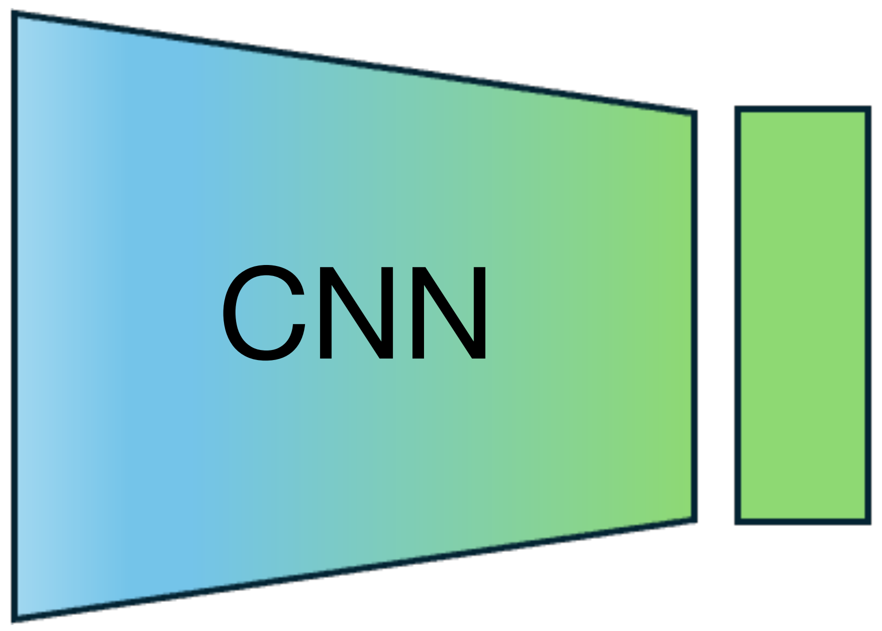
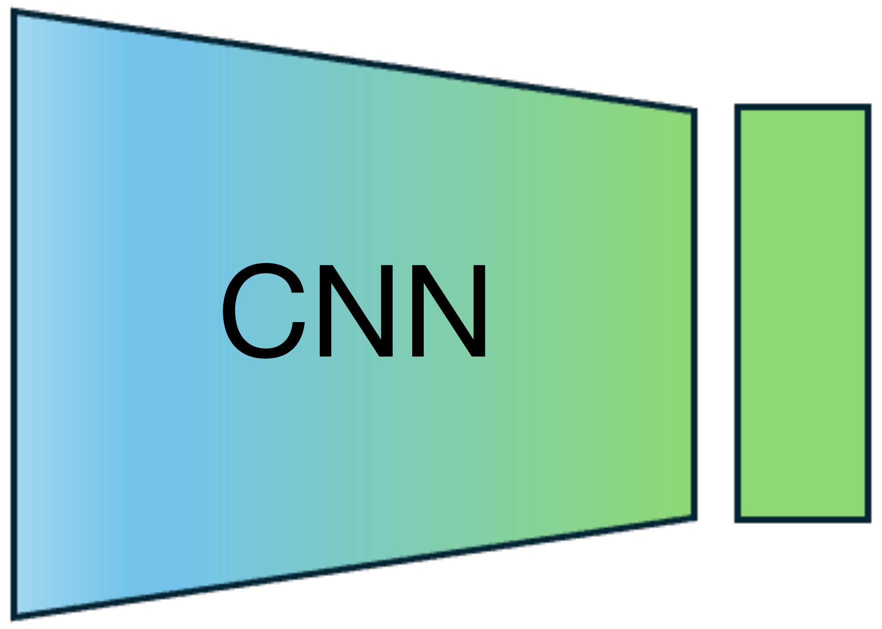
How do we interpret the outputs of a neural network trained on classification?

Yudi Xie, ICLR 2025

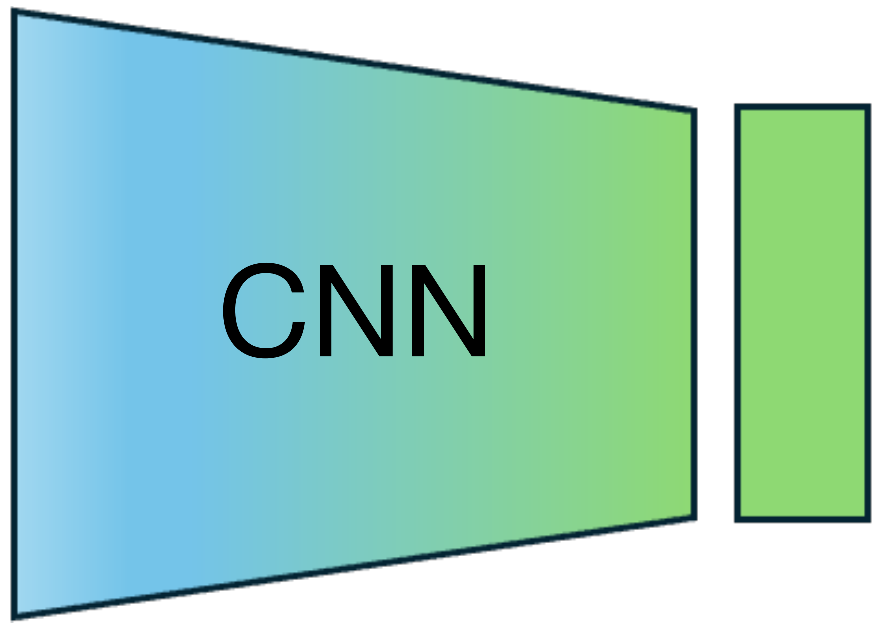
Training neural networks on classification tasks



Training neural networks on classification tasks

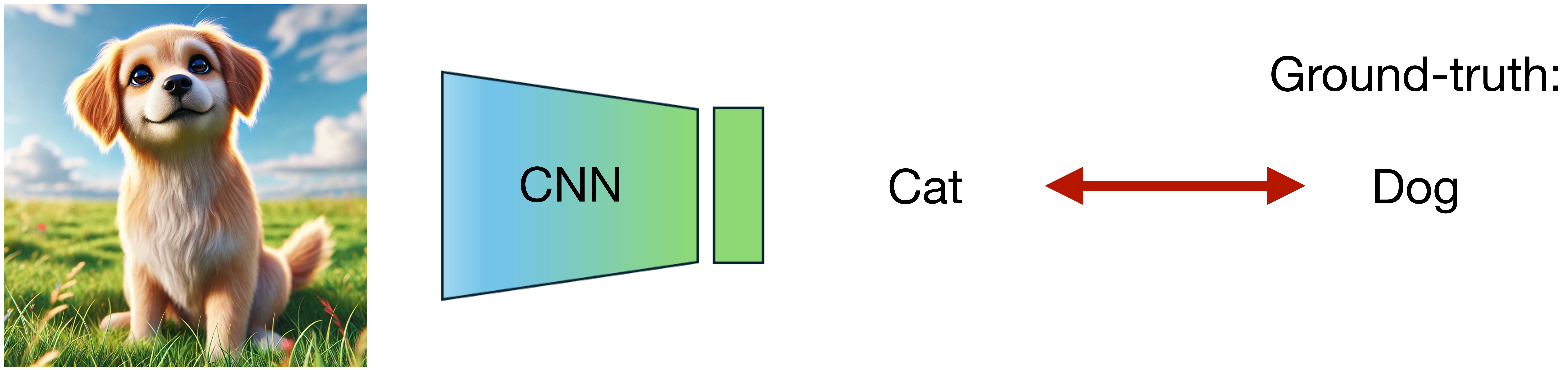


Training neural networks on classification tasks

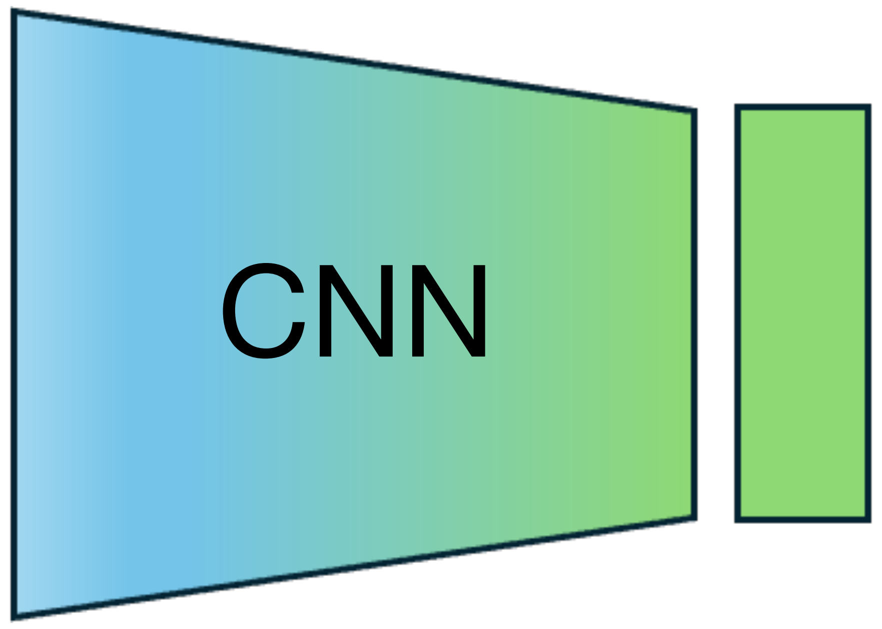


Cat

Training neural networks on classification tasks



Training neural networks on classification tasks



Ground-truth:

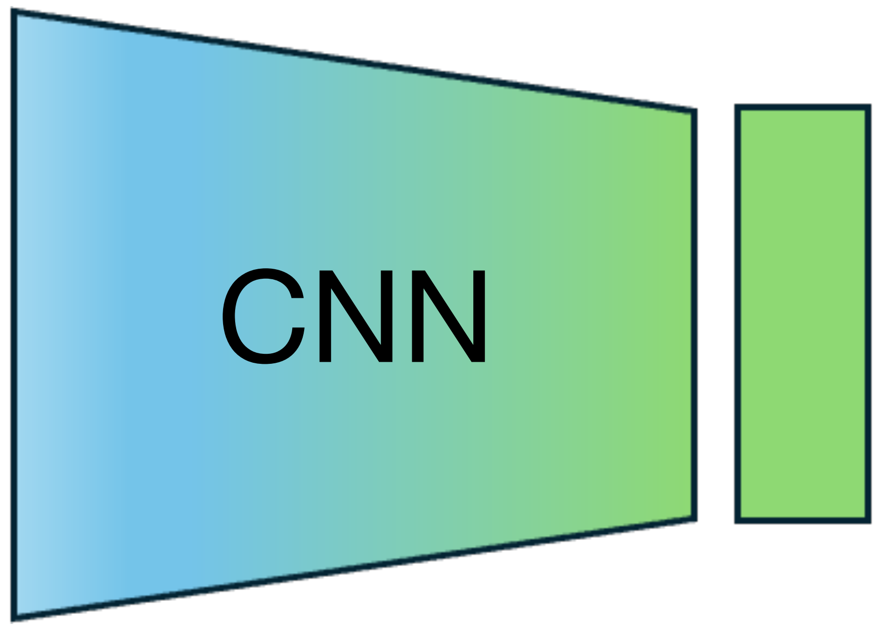
Cat ⟷ Dog

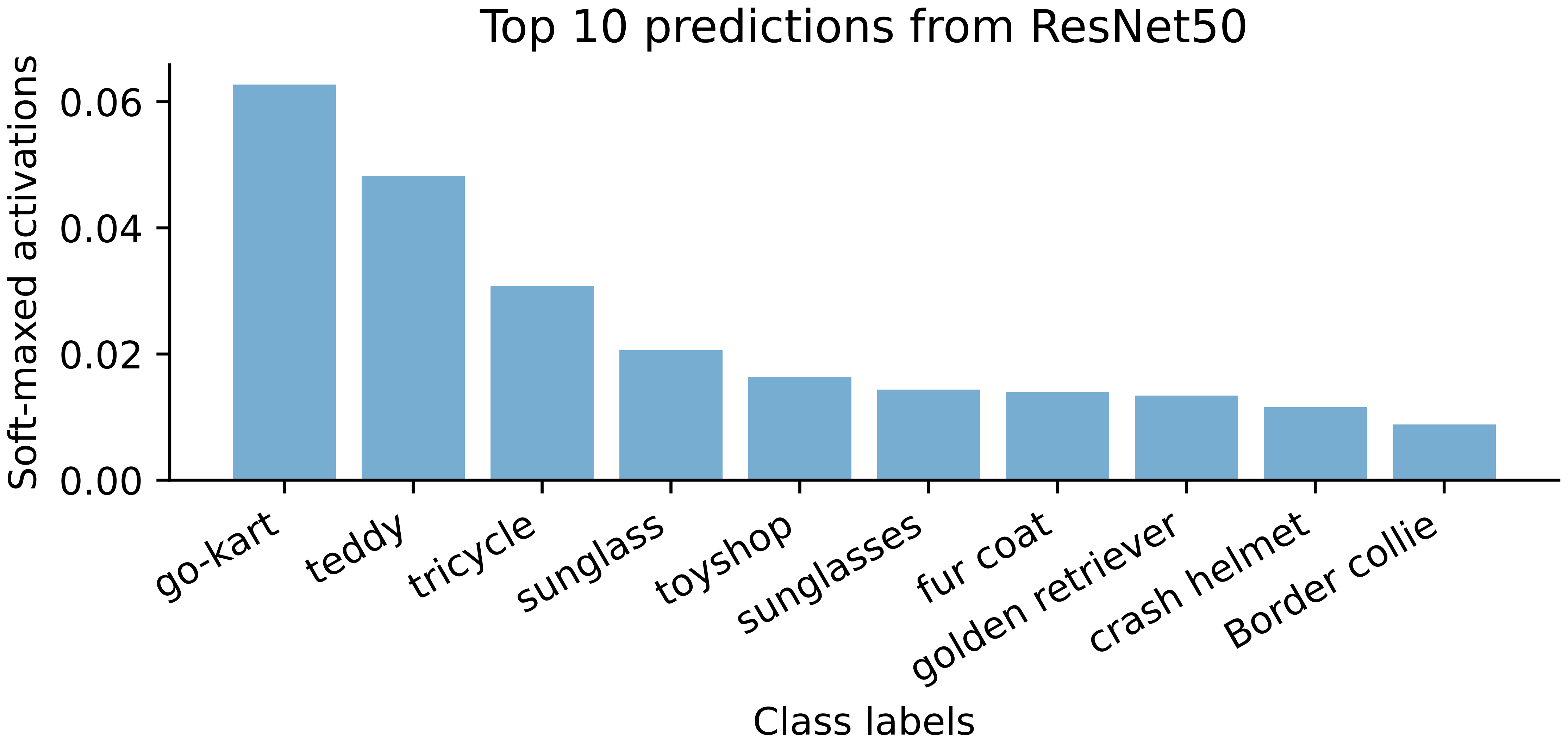
Cross-entropy loss (can be derived from maximum likelihood):

$$\mathbb{L} = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{M}\log(q_\theta(C = i|x_j)) \cdot 1\{c_j = i\}$$

Outputs of an ImageNet trained CNN

Outputs of an ImageNet trained CNN





What is the meaning of these output activations?

Minimizing cross-entropy is equivalent to minimizing KL divergence

Minimizing cross-entropy loss (can be derived from maximum likelihood):

$$\mathbb{L} = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{M}\log(q_\theta(C=i|x_j)) \cdot 1\{c_j = i\}$$

Minimizing cross-entropy is equivalent to minimizing KL divergence

Minimizing cross-entropy loss (can be derived from maximum likelihood):

$$\mathbb{L} = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{M}\log(q_\theta(C=i|x_j)) \cdot 1\{c_j = i\}$$

Is equivalent to …

Minimizing the KL-Divergence between the **Bayesian posterior** and the **model outputs**

$$\mathbb{L}_{KL}(P(C|X), q_\theta(C|X)) = \mathbb{E}_{x\sim P(X)} D_{KL}(P(C|x)|q_\theta(C|x))$$

Loss is minimized when outputs exactly match the posterior

We can derive a lower bound on the loss, and the above loss is minimized when the model outputs exactly match the posterior.

$$q_{\theta^*}(C = i|x) = P(C = i|x), \quad i \in \{1, \ldots, M\}$$

How to interpret the network outputs?

How to interpret the network outputs?

Training models using the cross-entropy loss pushes the outputs to match the Bayesian posterior P(C | X) of an ideal observer having access to the generative model P(X, C) that has generated the data.

How to interpret the network outputs?

Training models using the cross-entropy loss pushes the outputs to match the Bayesian posterior P(C | X) of an ideal observer having access to the generative model P(X, C) that has generated the data.

- When the generative model is **known**, the outputs should gradually match the Bayesian posterior P(C | X) during training.

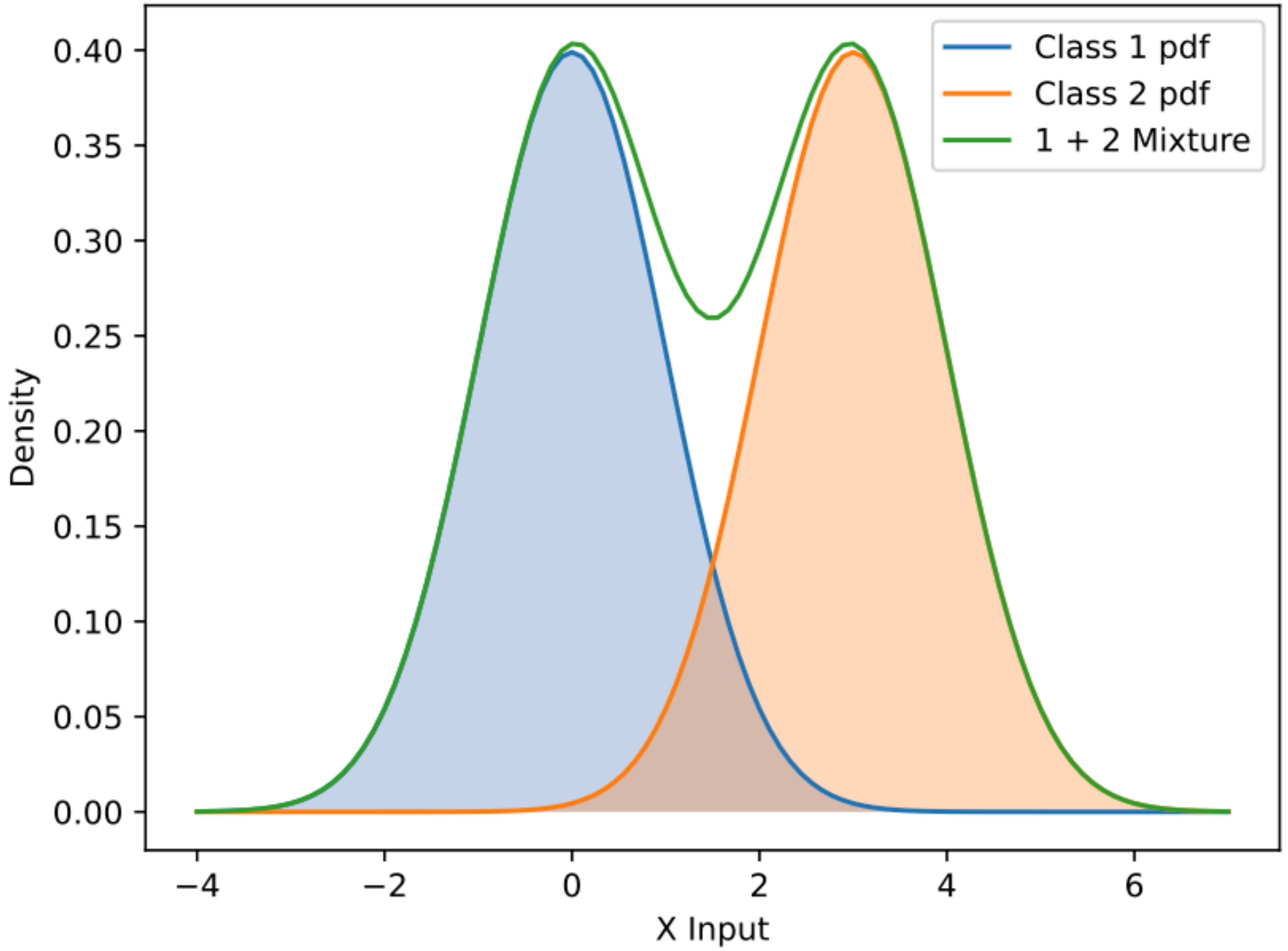
How to interpret the network outputs?

Training models using the cross-entropy loss pushes the outputs to match the Bayesian posterior P(C | X) of an ideal observer having access to the generative model P(X, C) that has generated the data.

- When the generative model is **known**, the outputs should gradually match the Bayesian posterior P(C | X) during training.
- When the generative model is **not known** (most real-world tasks), the outputs should match the Bayesian posterior calculated using a generative model of the data, if someone can find such a model.

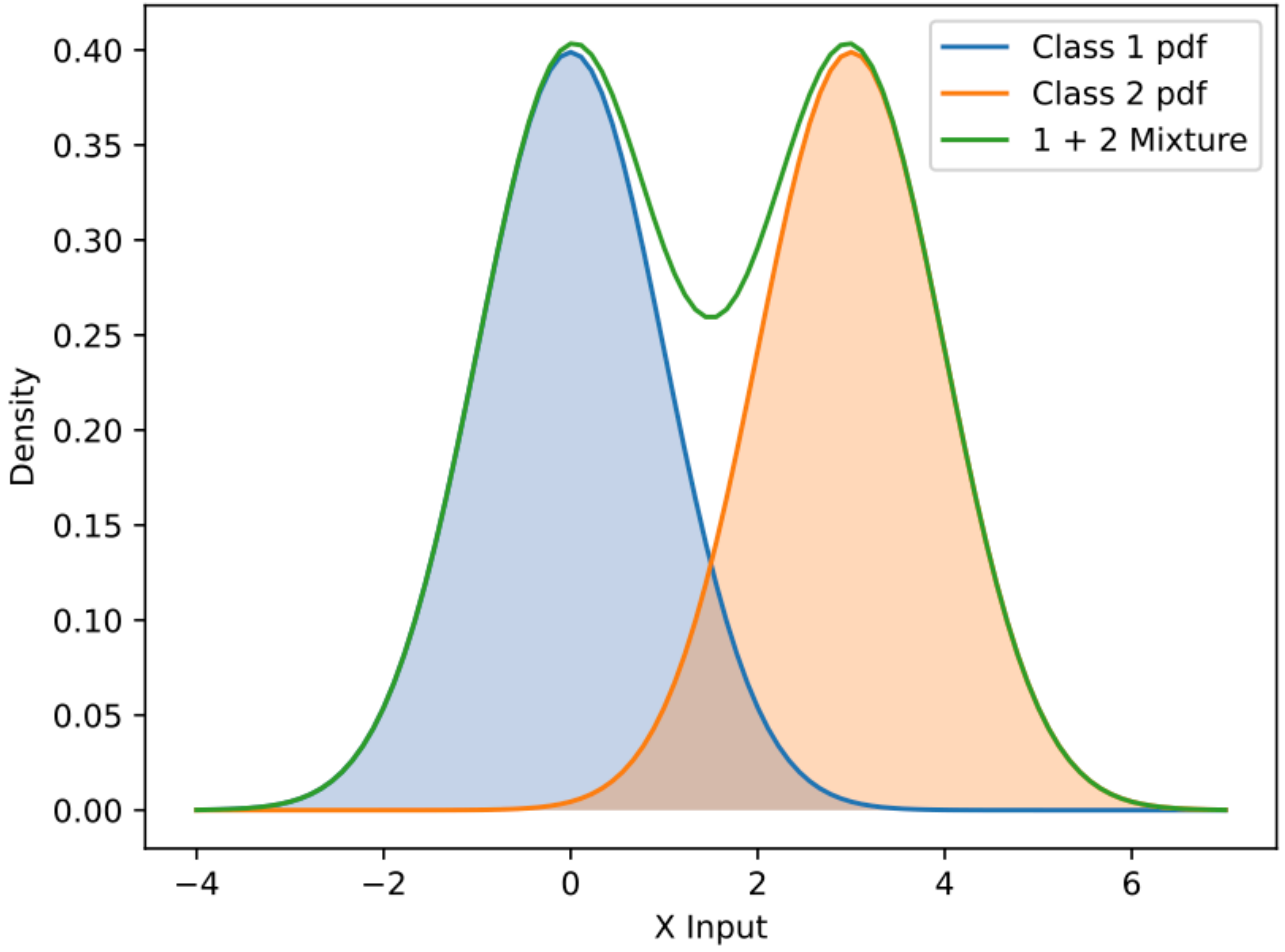
Empirical studies with known generative models: simple example

A simple classification example



Empirical studies with known generative models: simple example

A simple classification example

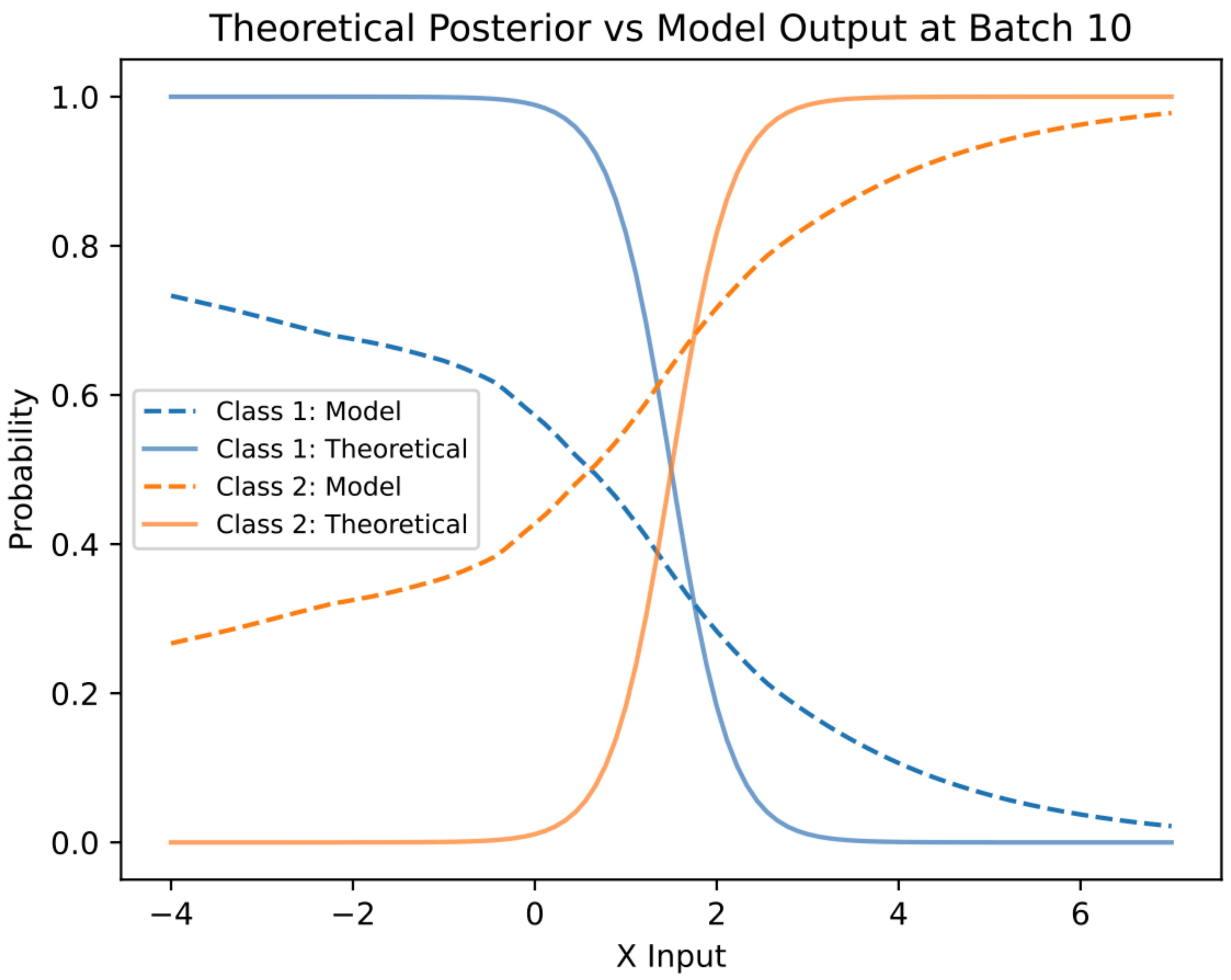


Derive the posterior using Bayes' rule

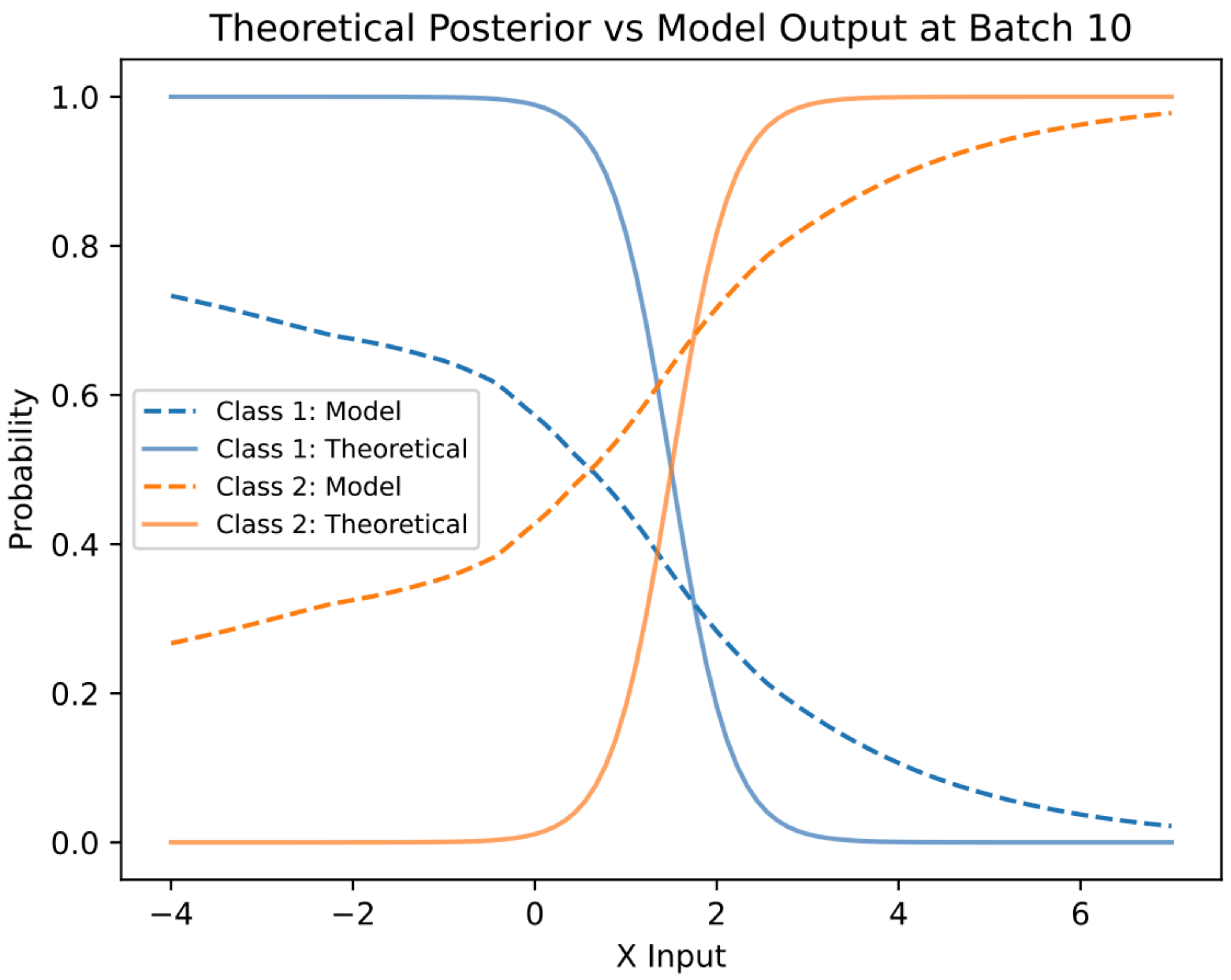
$$
\begin{aligned}
P(c_1|x) &= \frac{P(x|c_1)P(c_1)}{P(x)} = \frac{P(x|c_1)P(c_1)}{P(x|c_1)P(c_1) + P(x|c_2)P(c_2)} \\
&= \frac{1}{1 + \frac{\sigma_1}{\sigma_2} e^{\frac{\sigma_2^2(x-\mu_1)^2 - \sigma_1^2(x-\mu_2)^2}{2\sigma_1^2\sigma_2^2}}} \\
&= \frac{1}{1 + e^{\frac{6x-9}{2}}}
\end{aligned}
$$

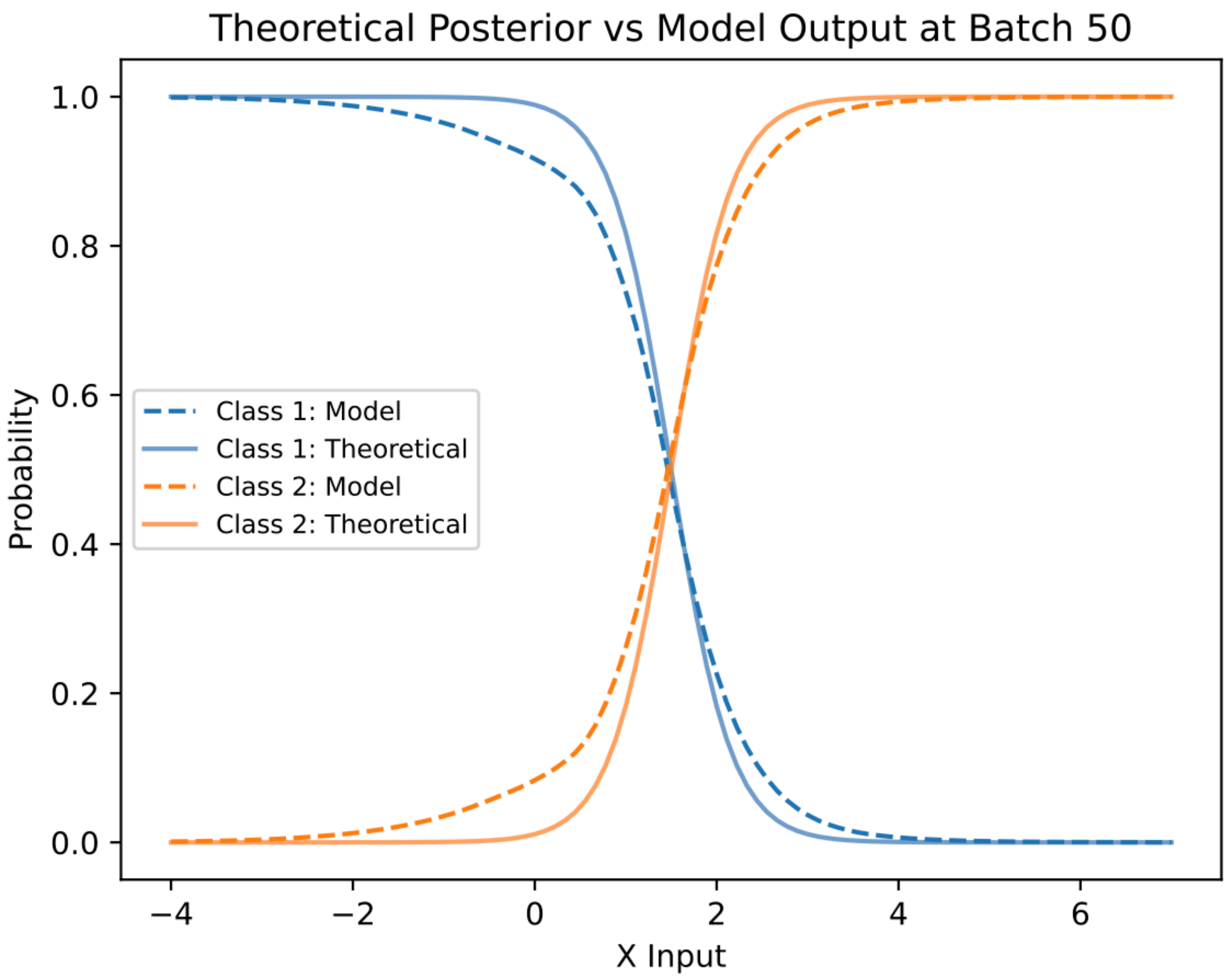
$$
\begin{aligned}
P(c_2|x) &= \frac{P(x|c_2)P(c_2)}{P(x)} = \frac{P(x|c_2)P(c_2)}{P(x|c_1)P(c_1) + P(x|c_2)P(c_2)} \\
&= \frac{1}{1 + \frac{\sigma_2}{\sigma_1} e^{\frac{\sigma_1^2(x-\mu_2)^2 - \sigma_2^2(x-\mu_1)^2}{2\sigma_1^2\sigma_2^2}}} \\
&= \frac{1}{1 + e^{\frac{-6x+9}{2}}}
\end{aligned}
$$

Empirical studies with known generative models: simple example

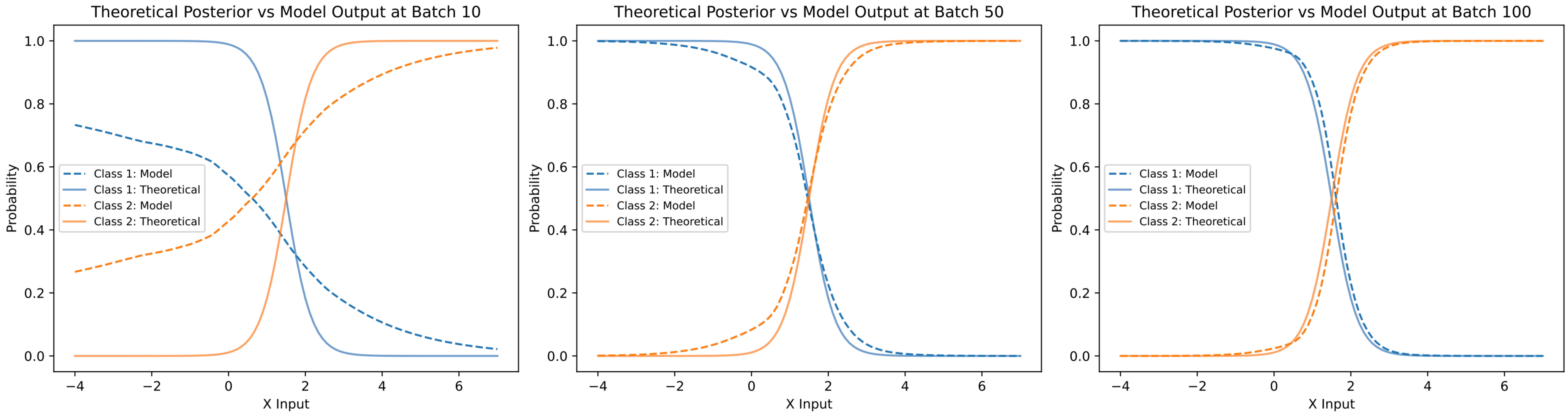


Empirical studies with known generative models: simple example

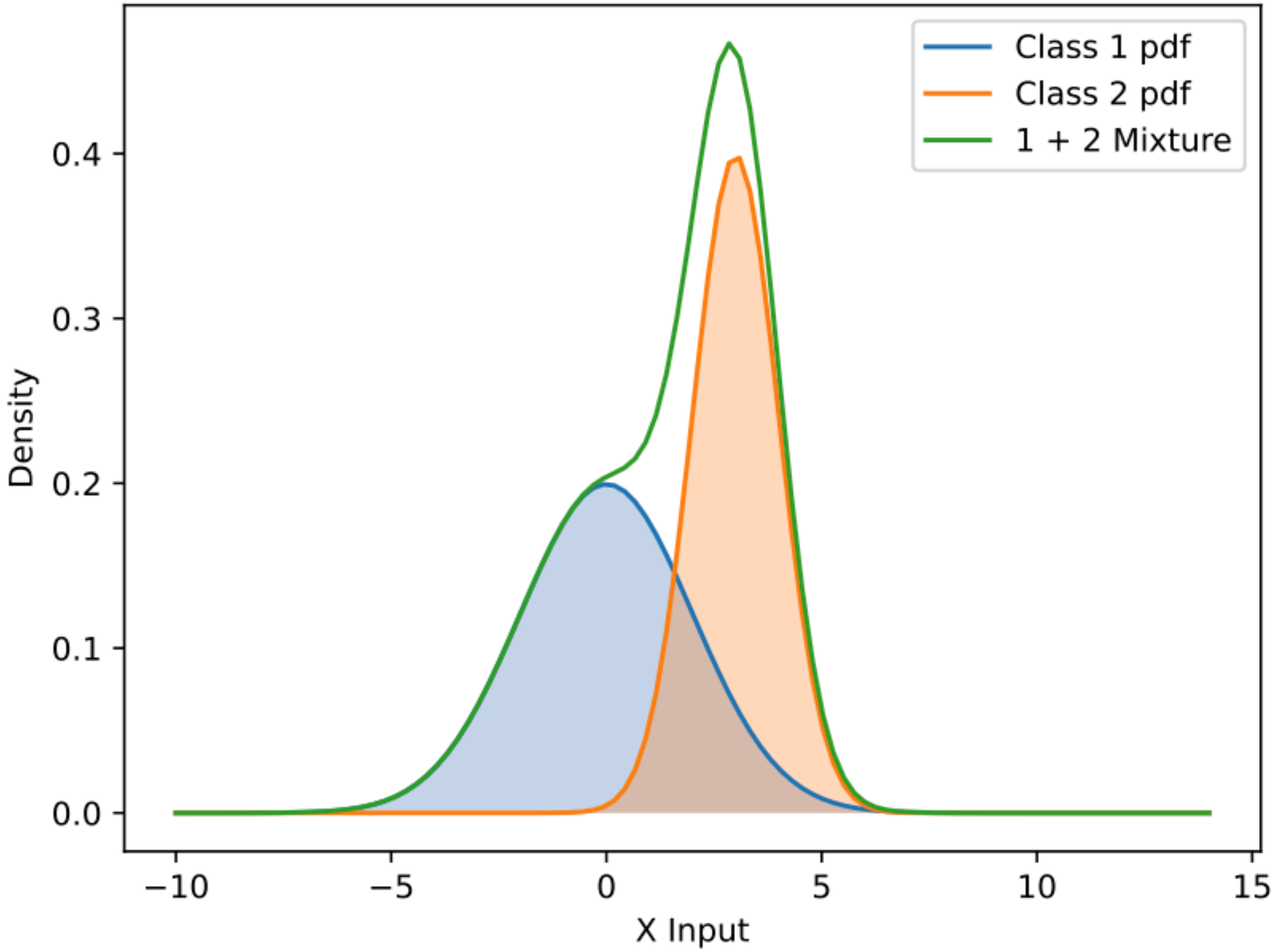




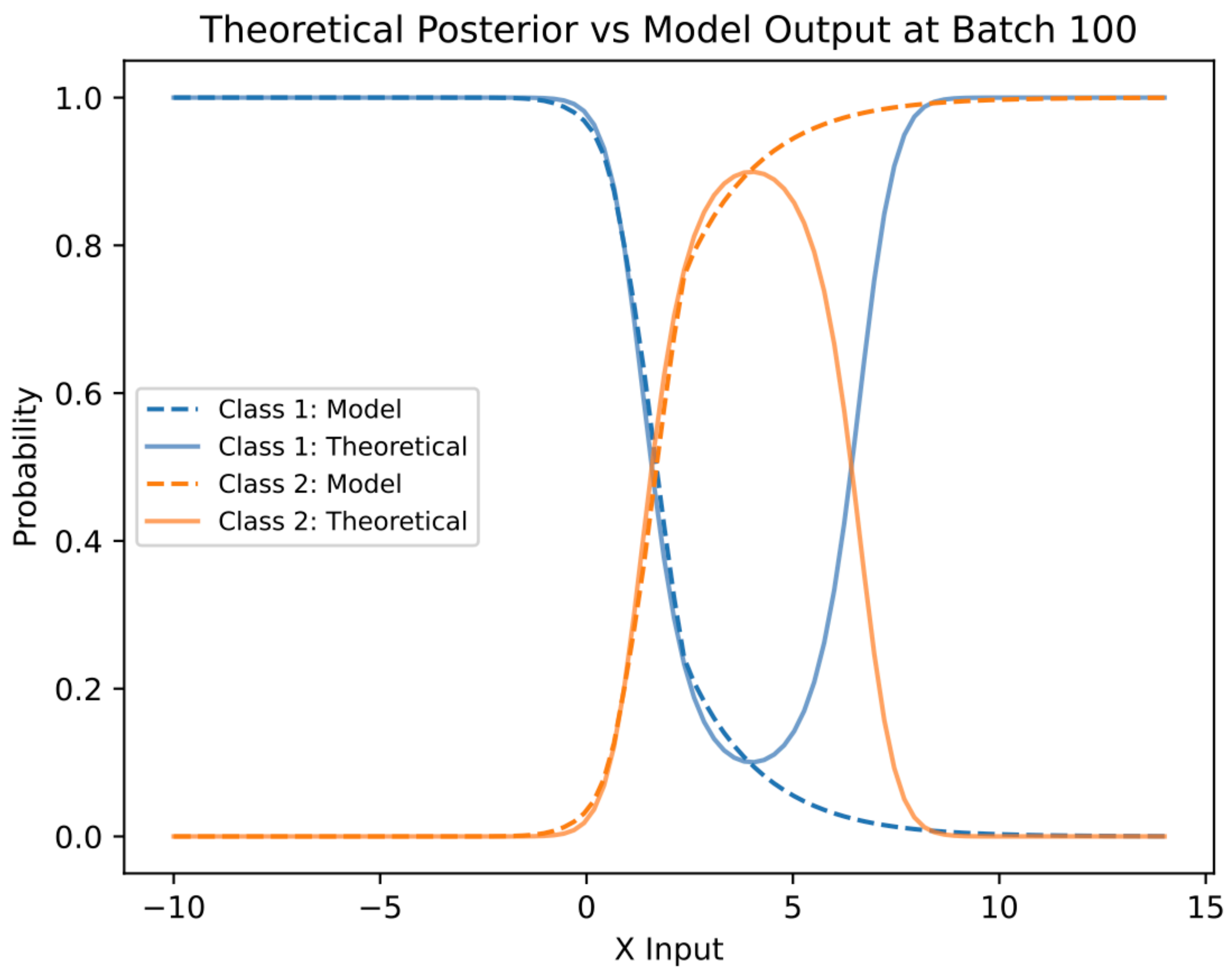
Empirical studies with known generative models: simple example



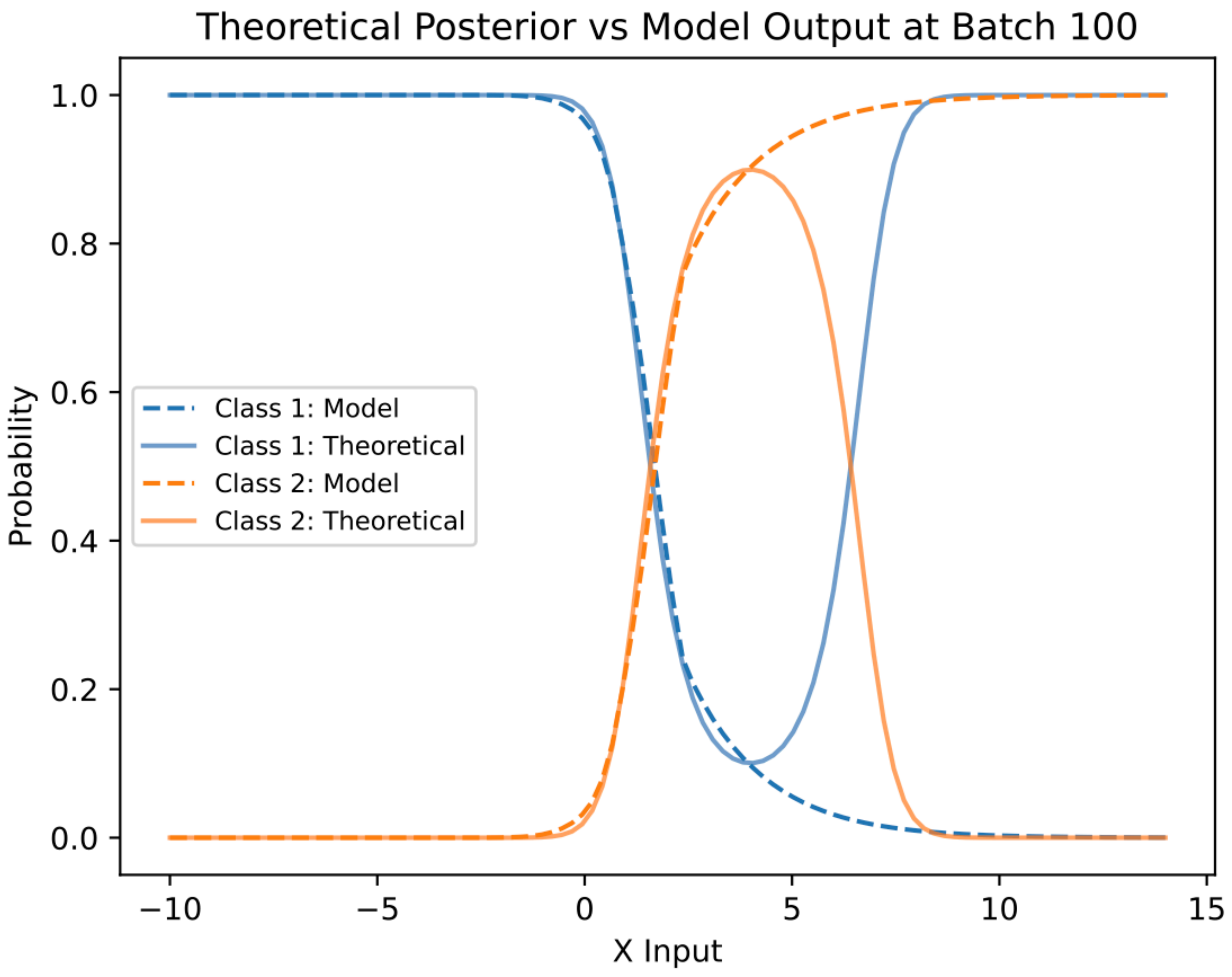
Empirical studies with known generative models: more complex example

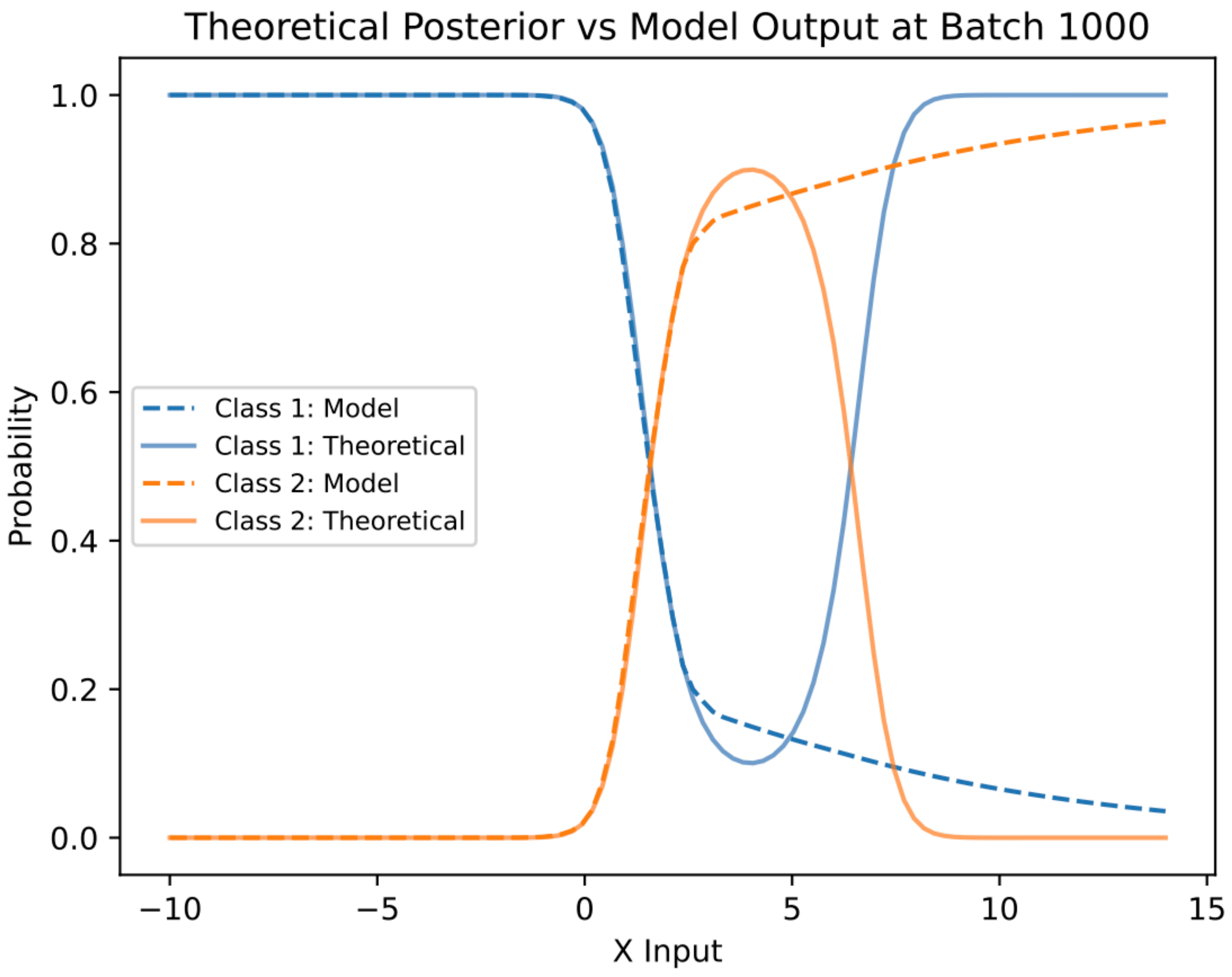


Empirical studies with known generative models: more complex example

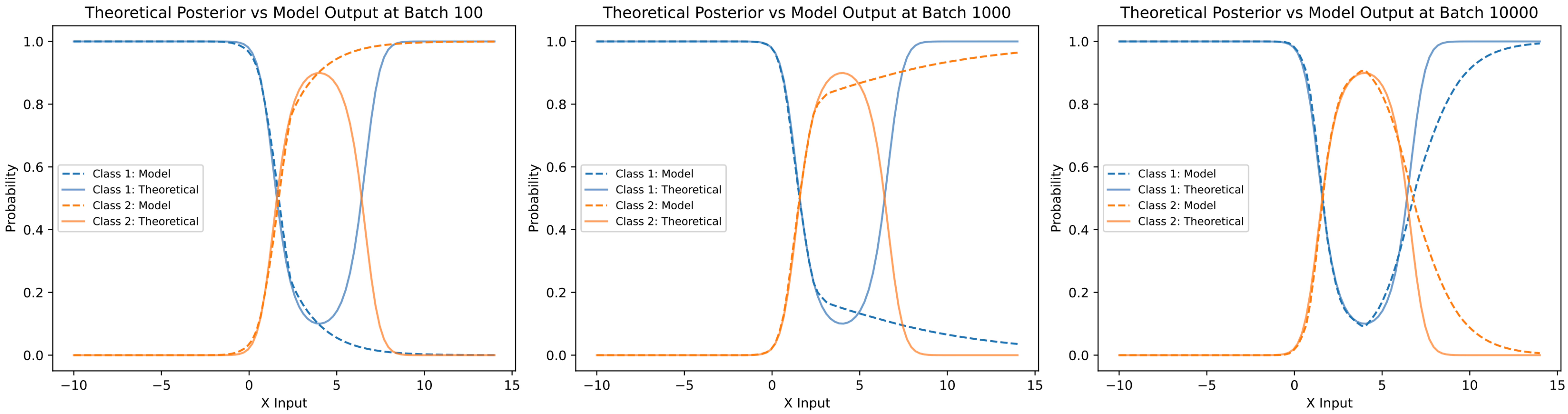


Empirical studies with known generative models: more complex example





Empirical studies with known generative models: more complex example



Summary

Summary

- Training neural network using cross-entropy loss pushes the outputs of the model to match the Bayesian posterior calculated using a generative model that has generated the data.

Summary

- Training neural network using cross-entropy loss pushes the outputs of the model to match the Bayesian posterior calculated using a generative model that has generated the data.
- How well the model outputs actually approximate the posterior could depend on multiple factors, such as the shape of the posterior, the generative distribution, and model training details.