Section 3 Report

Juntao DONG

February 2, 2026

1 Flow Matching Schedules

1.1 Summary

In this lab session, we explored the Flow Matching framework both theoretically and experimentally. We studied different schedules, in particular the linear and cosine designs, and observed how their theoretical properties, such as variance preservation, manifest in practice. Through simple 2D experiments and image generation tasks, we saw that the cosine schedule leads to more stable trajectories, while the linear schedule can distort the geometry of the data. Overall, this lab provided an intuitive and hands-on understanding of Flow Matching and highlighted why diffusion-based methods are effective for generative modeling.

1.2 Questions and Answers

1. **A popular schedule of Diffusion Models/Flow Matching is the "cosine schedule". This is of the form**

   $$\alpha_t = \cos(at),\ \beta_t = \sin(bt).$$

   Determine the smallest a and b such that the border conditions are respected;

   Answer. The border conditions are

   $$\alpha_0 = 1, \quad \beta_0 = 0, \quad \alpha_1 = 0, \quad \beta_1 = 1.$$

   The first two are automatically satisfied. The remaining conditions give

   $$\cos(a) = 0, \qquad \sin(b) = 1.$$

   The smallest positive solutions are

   $$a = \frac{\pi}{2}, \qquad b = \frac{\pi}{2}.$$

2. **Calculate $\dfrac{dX_t}{dt}$ in the case of the cosine schedule;**

   Answer. For the cosine schedule with $a = b = \frac{\pi}{2}$,

   $$X_t = \cos\left(\frac{\pi t}{2}\right) X_0 + \sin\left(\frac{\pi t}{2}\right) X_1.$$

   Differentiating with respect to t yields

   $$\frac{dX_t}{dt} = -\frac{\pi}{2}\sin\left(\frac{\pi t}{2}\right) X_0 + \frac{\pi}{2}\cos\left(\frac{\pi t}{2}\right) X_1.$$

3. **Consider the linear schedule between two Gaussian random variables:**

   $$X_t = (1-t)X_0 + tX_1,$$

   with $X_0 \sim \mathcal{N}(0,1)$ and $X_1 \sim \mathcal{N}(1,1)$. By looking at the case $t = \frac{1}{2}$, show that the schedule is not variance preserving;

Answer. At $t = \frac{1}{2}$,

$$X_{1/2} = \frac{1}{2}X_0 + \frac{1}{2}X_1.$$

Since X_0 and X_1 are independent,

$$\mathrm{Var}(X_{1/2}) = \frac{1}{4}\,\mathrm{Var}(X_0) + \frac{1}{4}\,\mathrm{Var}(X_1) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2} \neq 1.$$

Therefore, the linear schedule is not variance preserving.

4. **Show that the cosine schedule is variance preserving (using the same technique);**

Answer. With the cosine schedule,

$$X_t = \cos\left(\frac{\pi t}{2}\right) X_0 + \sin\left(\frac{\pi t}{2}\right) X_1.$$

Since X_0 and X_1 are independent and both have variance 1,

$$\mathrm{Var}(X_t) = \cos^2\left(\frac{\pi t}{2}\right) + \sin^2\left(\frac{\pi t}{2}\right) = 1,$$

for all $t \in [0, 1]$. Hence, the cosine schedule is variance preserving.

5. $\star$**Why is it an advantage to be variance preserving (think about the geometry of the generation process) ?**

Answer. Variance preservation maintains a constant scale of the data distribution throughout the generation process. Geometrically, this avoids artificial contraction or expansion of the particle cloud, leading to smoother trajectories and a more stable approximation of the velocity field. This improves both numerical stability and sample quality.

6. **Do the experimental observations confirm your theoretical predictions concerning the variance preservation of the schedules ? To be sure about this, you can calculate the variance of the generated data at each time step.**

Answer. Yes, the experiments confirm the theoretical predictions: the linear schedule is not variance preserving, whereas the cosine schedule is variance preserving, as expected from the analysis in the Gaussian case.

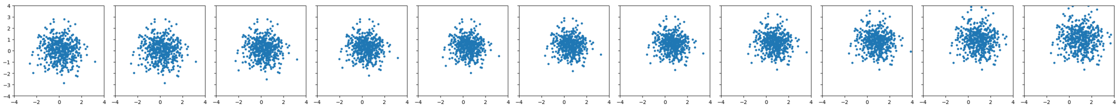

Figure 1: Linear Desgin

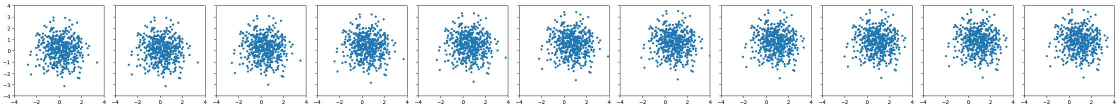

Figure 2: Cosine Desgin

7. **What is the major advantage of Diffusion Models/Flow Matching over Generative Adversarial Networks (GANs) ?**

Answer. The major advantage of Diffusion Models and Flow Matching over GANs is their training stability and their ability to better cover the data distribution. Unlike GANs, which rely on an adversarial minimax optimization that can be unstable and prone to mode collapse, Diffusion Models and Flow Matching optimize a direct regression or likelihood-related objective. This leads to more stable training and more reliable generation of diverse samples.

8. **What is the computational disadvantage of Diffusion Models/Flow Matching in comparison to GANs ?**

   Answer. The main computational disadvantage of Diffusion Models and Flow Matching is that sample generation requires many iterative steps. Each sample is produced by evaluating the neural network multiple times along a discretized time trajectory, whereas GANs generate samples in a single forward pass of the generator.

9. $\star$ **GANs generate images in one "step", ie with one nerual network, whereas Flow Matching does this in several steps. In light of this, if we allow each method to have a fixed number of neural network parameters, why do you think the Flow Matching algorithm produces better results ?**

   Answer. Flow Matching produces better results because it decomposes a complex generative task into a sequence of simpler transformations. Each step only needs to model a small and local change in the data space, rather than learning a highly complex mapping in a single step. This gradual transport of probability mass provides better geometric control of the generation process and allows errors to be corrected over multiple steps, which leads to higher sample quality and better coverage of the target distribution.

2 Bayesian Linear Regression and NN

2.1 Summary

In this lab, we studied Bayesian approaches to regression and classification and their ability to quantify uncertainty. We first analyzed Bayesian Linear Regression, deriving the posterior and predictive distributions and observing how uncertainty depends on data availability and model assumptions. We then extended these ideas to non-linear models and neural networks using approximate inference methods, highlighting the limitations of point estimates and the benefits of probabilistic modeling. Overall, this lab illustrated how Bayesian methods provide more informative predictions by explicitly modeling uncertainty, especially in regions with limited or ambiguous data.

2.2 Questions and Answers

2.2.1 Linear Model

1. **Recall closed form of the posterior distribution in linear case. Then, code and visualize posterior sampling. What can you observe?**

   Answer. Assuming a Gaussian prior $p(w) = \mathcal{N}(0, \alpha^{-1}I)$ and a Gaussian likelihood $p(y|x, w) = \mathcal{N}(\Phi^\top w, \beta^{-1})$, the posterior distribution over the weights remains Gaussian:

   $$p(w|X, Y) = \mathcal{N}(\mu, \Sigma),$$

   with

   $$\Sigma^{-1} = \alpha I + \beta \Phi^\top \Phi, \qquad \mu = \beta \Sigma \Phi^\top Y.$$

   By sampling from this posterior distribution, we observe that for a small number of training points, the posterior variance is large, reflecting high epistemic uncertainty. As the number of training examples increases, the posterior distribution concentrates around the ground truth parameters and its variance decreases, indicating increased confidence in the model.

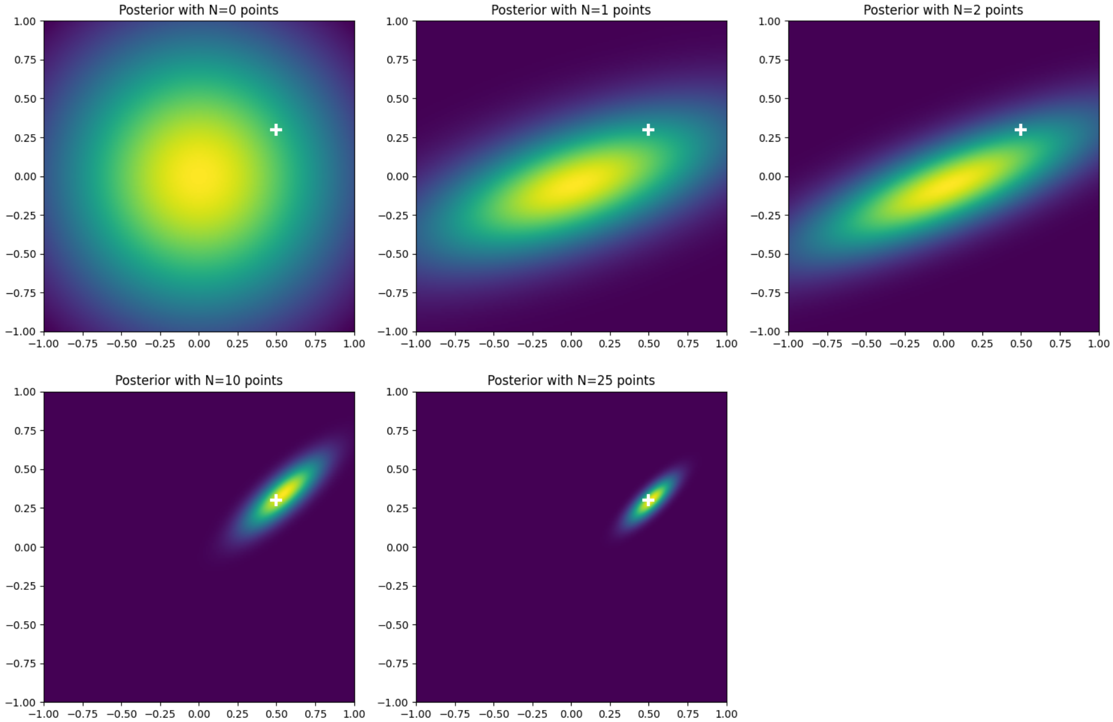

Figure 3: Posterior Distribution

2. **Recall and code the closed form of the predictive distribution in the linear case.**

   Answer. The predictive distribution for a new input $x^\star$ is obtained by marginalizing over the posterior distribution of the weights:

$$p(y|x^\star, D) = \int p(y|x^\star, w)p(w|D)\,dw.$$

   Since both terms are Gaussian, the predictive distribution is also Gaussian:

$$p(y|x^\star; D, \alpha, \beta) = \mathcal{N}\big(\mu^\top \Phi(x^\star),\ \beta^{-1} + \Phi(x^\star)^\top \Sigma \Phi(x^\star)\big)\,.$$

   The predictive variance naturally decomposes into an aleatoric term β^{-1} and an epistemic term $\Phi(x^\star)^\top \Sigma \Phi(x^\star)$.

3. **Based on previously defined `f_pred()`, predict on the test dataset and visualize the results.**

   Answer. The posterior mean provides a good fit to the data in the region covered by the training points. The predictive uncertainty is minimal near the training data and increases as we move away from this region. This behavior is illustrated by widening confidence intervals outside the training domain.

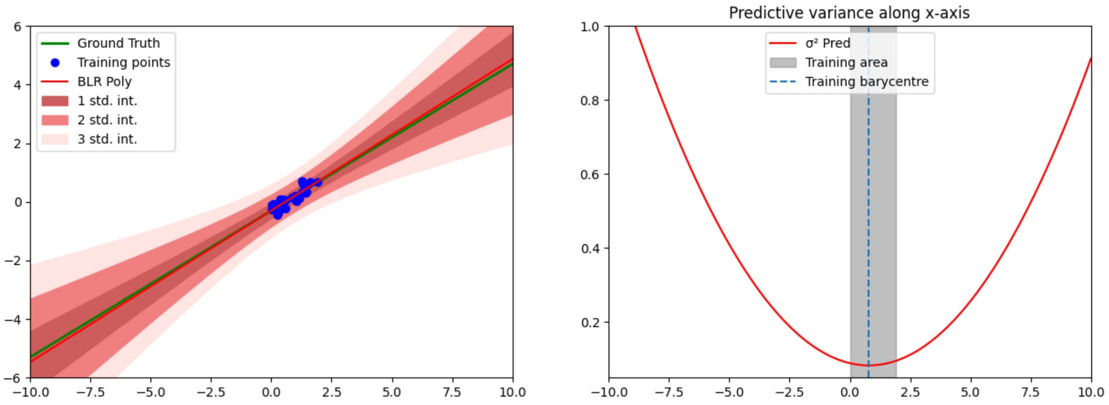

Figure 4

4. **Analyse these results. Why does the predictive variance increase far from the training distribution? Prove it analytically in the case where $\alpha = 0$ and $\beta = 1$.**

Answer. Empirically, the predictive variance is smallest in regions where training data are present and increases as the test point moves away from the training distribution, reflecting increased epistemic uncertainty.

Analytically, when $\alpha = 0$ and $\beta = 1$, the predictive variance simplifies to

$$\sigma^2_{\text{pred}}(x^\star) = \Phi(x^\star)^\top \Sigma \Phi(x^\star),$$

which can be shown to be proportional to

$$\sum_{i=1}^{N} (x_i - x^\star)^2.$$

As $x^\star$ moves farther from the training points, this quantity increases, explaining the growth of predictive variance.

5. **Bonus: What happens when applying Bayesian Linear Regression on a dataset with a "hole"?**

Answer. In the presence of a "hole" in the data distribution, the predictive variance can be unexpectedly low at the barycenter of the dataset, even though no training points are present there. This occurs because the barycenter lies between clusters of data points, leading to smaller average squared distances. This phenomenon highlights the influence of the model assumptions and prior, which may result in overconfident predictions in certain regions.

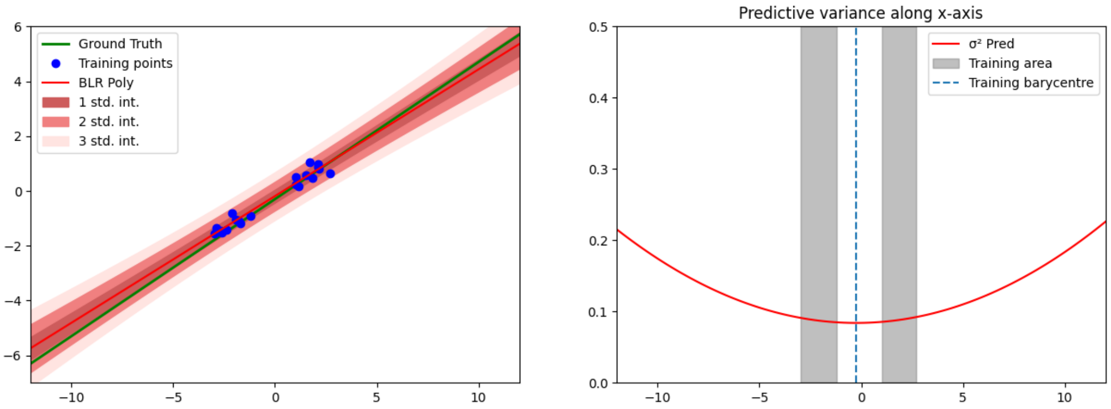

Figure 5

2.2.2 Non Linear Models

1. **Code and visualize results on sinusoidal dataset using polynomial basis functions. What can you say about the predictive variance?**

   Answer. Using polynomial basis functions allows the Bayesian regression model to capture more complex, non-linear patterns compared to the linear case. On the sinusoidal dataset, the model is able to fit the training data reasonably well within the observed region.

   From the visualizations, we observe that the predictive variance is small in regions where training data are present and increases as we move away from them. In particular, the uncertainty grows rapidly outside the training interval, indicating high epistemic uncertainty due to lack of data support. Although the polynomial basis enables greater flexibility, the model still struggles to extrapolate the periodic structure of the sinusoidal function beyond the training region.

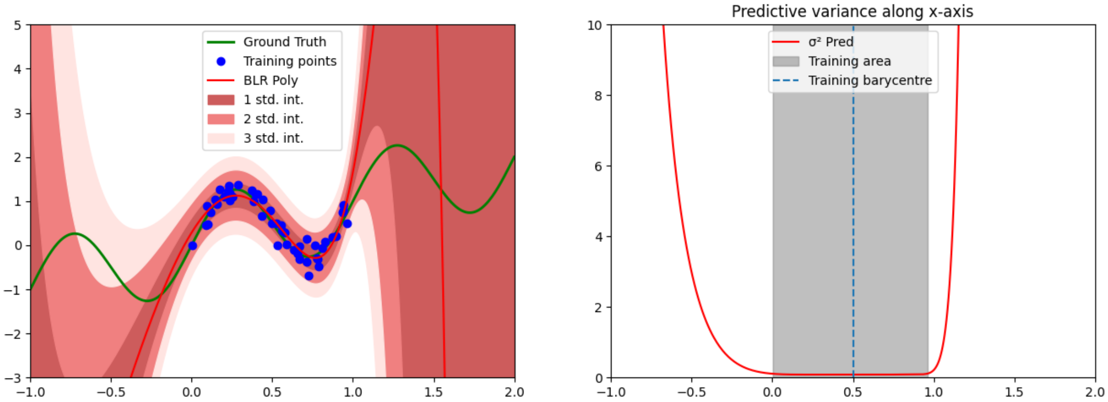

Figure 6

3 Uncertainty Applications

3.1 Summary

In this lab, we explored uncertainty estimation in machine learning models from both a theoretical and practical perspective. Starting with Bayesian Linear Regression, we analyzed posterior and predictive distributions and observed how epistemic uncertainty evolves with data availability. We then extended these ideas to classification tasks using approximate inference methods such as Laplace approximation, variational inference, and Monte-Carlo Dropout. Finally, through practical experiments on MNIST, we showed how uncertainty measures can be used to analyze ambiguous samples and improve failure prediction. Overall, this lab provided a concrete understanding of why uncertainty estimation is crucial for reliable machine learning models beyond raw predictive accuracy.

3.2 Questions and Answers

3.2.1 Monte-Carlo Dropout on MNIST

1. **What can you say about the images themselves? How do the histograms along them help to explain failure cases? Finally, how do probability distributions of random images compare to the previous top uncertain images?**

   Answer. The most uncertain images identified by the variation ratio are visually ambiguous and often difficult to classify even for a human observer. These images typically present unclear digit shapes, overlaps, or unusual writing styles, which naturally lead to inconsistent predictions across stochastic forward passes.

The associated histograms provide insight into these failure cases by showing a large dispersion of output probabilities across classes. Unlike confident predictions, where the probability mass is concentrated on a single class, uncertain images exhibit spread-out distributions and frequent changes in the predicted class. This indicates strong epistemic uncertainty captured by Monte-Carlo Dropout.

In comparison, random images from the test set usually display more concentrated probability distributions, even when misclassified. Their histograms tend to be less dispersed than those of the most uncertain images, suggesting that high uncertainty is not merely due to misclassification, but rather to intrinsic ambiguity in the input data.

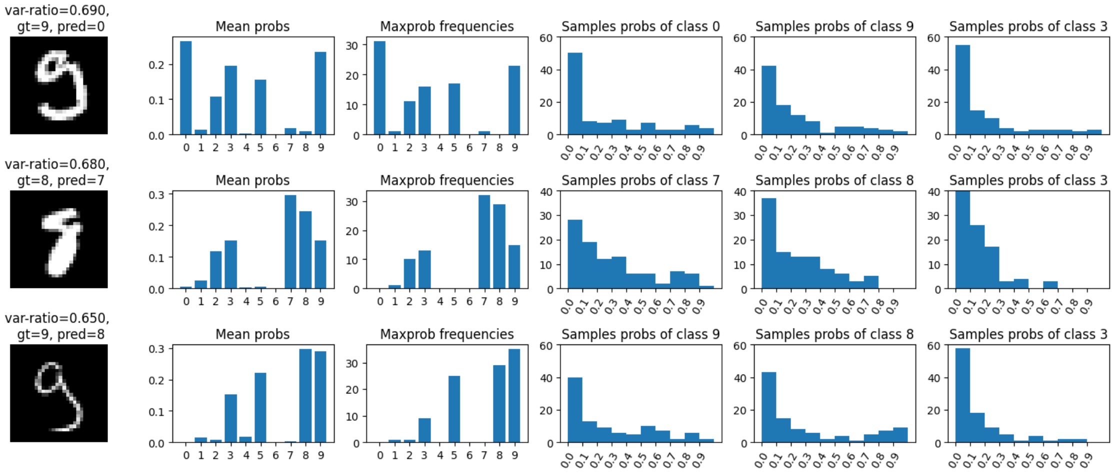

Figure 7

3.2.2 Failure Prediction

2. **Why is Maximum Class Probability (MCP) not a good metric for failure prediction? What alternative confidence measure can be used instead, and why?**

Answer. Maximum Class Probability (MCP) is not a reliable metric for failure prediction because modern neural networks tend to be overconfident. As a result, both correct and incorrect predictions can be associated with high MCP values, leading to significant overlap between success and failure distributions.

An alternative confidence measure is the True Class Probability (TCP), which corresponds to the probability assigned to the ground-truth class. When a model makes a mistake, the TCP is usually low, making it a more discriminative indicator of failure. However, since the true label is not available at test time, approaches such as ConfidNet aim to directly regress the TCP from the input data.

These methods provide a more reliable estimate of uncertainty and allow better separation between correct and incorrect predictions, making them well suited for failure prediction tasks.

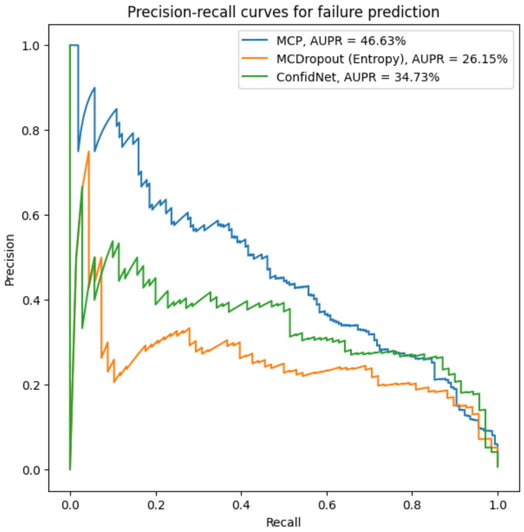
Precision-recall curves for failure prediction
MCP, AUPR = 46.63%
MCDropout (Entropy), AUPR = 26.15%
ConfidNet, AUPR = 34.73%
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

Figure 8