Weekly Study Report

Wang Ma

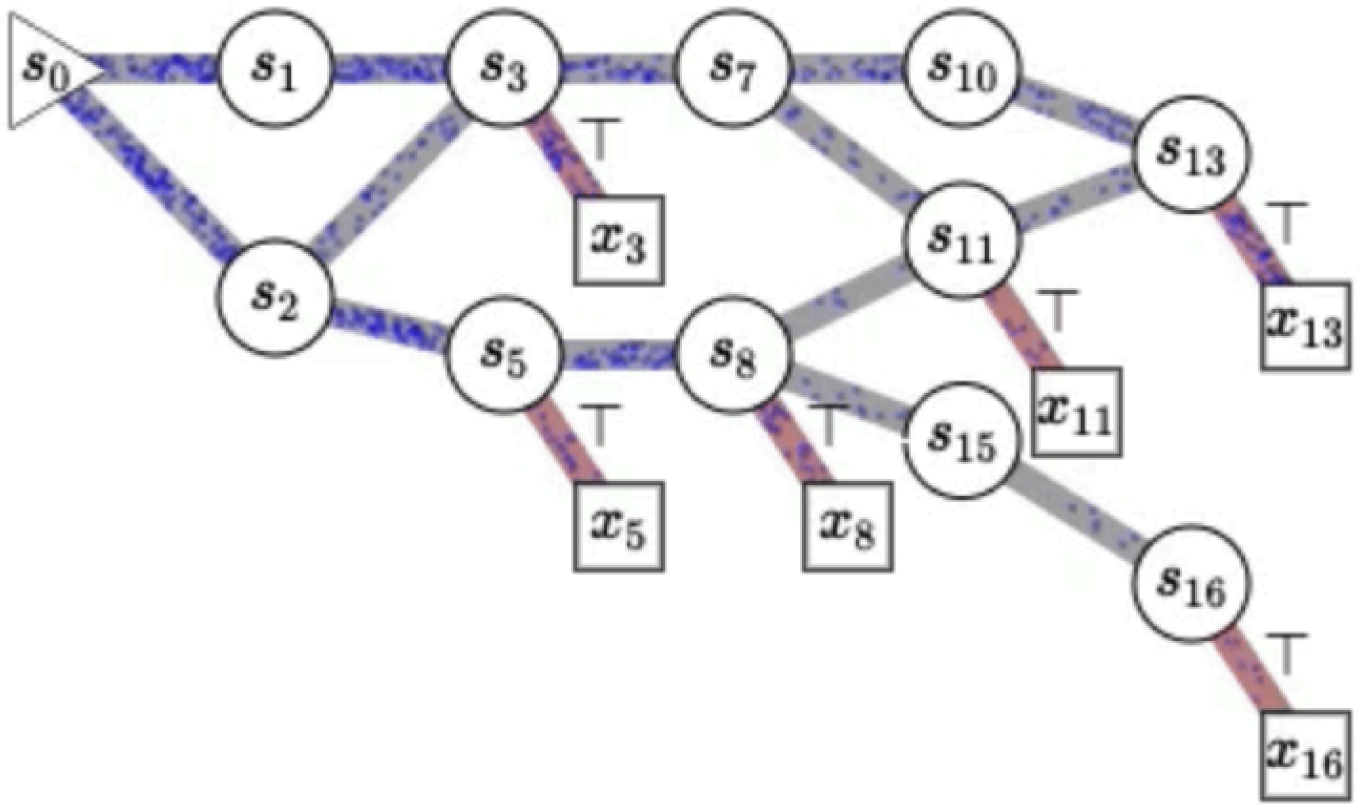
2024-10-11

Electrical, Computer, and Systems Engineering Department
Rensselaer Polytechnic Institute

Outline

1. DAG-GFlowNet

1.1.1 The Structure



1.1.2 Flow-matching Condition

The Goal of a GFlowNet is to find a *flow* that satisfies, for all states $s \in S$, the following ***flow-matching condition:***

$$\sum_{s \in Pa(s')} F_\theta(s \to s') - \sum_{s'' \in Ch(s')} F_\theta(s' \to s'') = R(s'),$$

where $F_\theta(s \to s') \geq 0$ is a scalar representing the flow from state s to s', typically parametrized by a neural network. The ***reward*** $R(s) \geq 0$ indicates a notion of "preference" for certain states.

A GFlowNet induces a generative process to sample complete states with probability:

$$P(s) \propto R(s),$$

and a transition probability along a sampled trajectory:

$$p(s_{t+1} \mid s_t) \propto F_\theta(s_t \to s_{t+1}).$$

1.1.3 Detailed-balance Condition

Since the flows are added together, one of the downsides of the flow-matching condition is that **flows tend to be orders of magnitude larger the closer we are of the initial state** , making it challenging to parametrize F_θ, e.g.,

$$F_\theta(s_1 \to s_2) = 0.8, \quad F_\theta(s_{T-1} \to s_T) = 0.00001,$$

the model F_θ need to correctly accommodate this different scales of output.

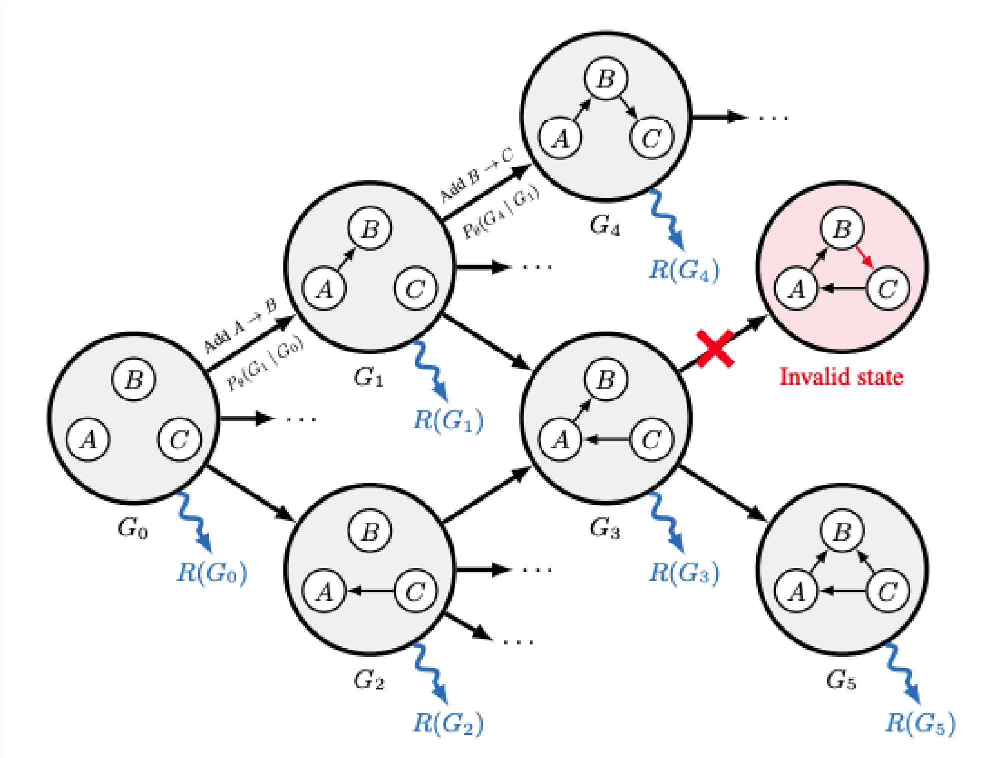
The ***detailed-balance condition*** for all transitions $s \to s'$, where s_f is the terminate state.:

$$R(s')P_B(s|s')P_\theta(s_f\ |s) = R(s)P_\theta(s'|s)P_\theta(s_f|s').$$

The ***detailed-balance loss***:

$$L(\theta) = \sum_{s \to s'} \left[\log \frac{R(s')P_B(s|s')P_\theta(s_f\ |s)}{R(s)P_\theta(s'|s)P_\theta(s_f|s')}\right]^2.$$

1.2.1 The Structure



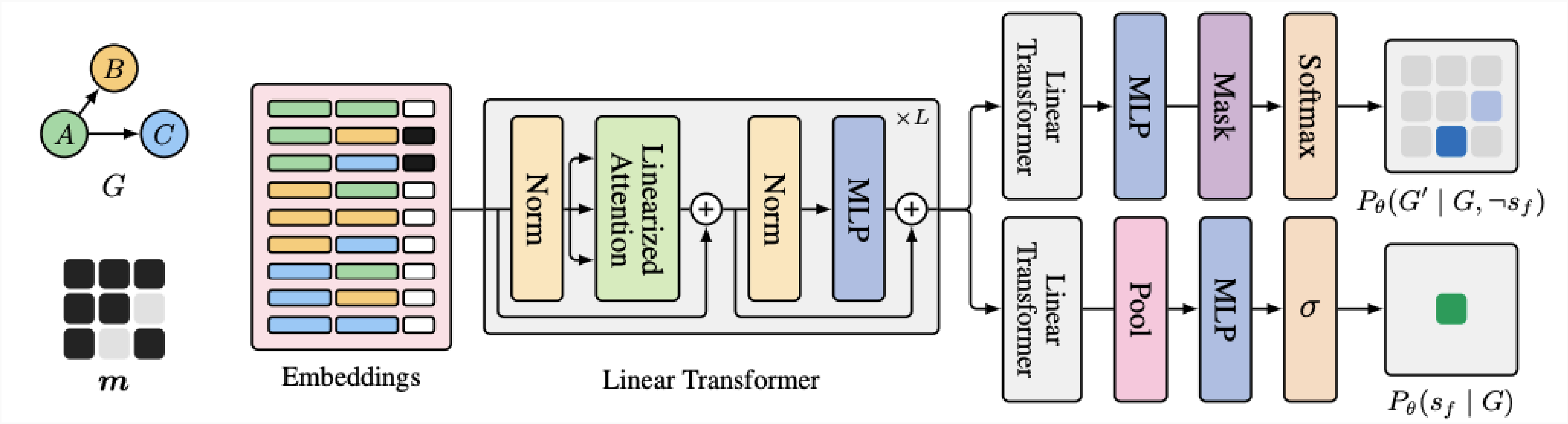
The Probability of taking a transition $G \to G'$ is given by

$$P_\theta(G'|G) = \left(1 - P_\theta\left(s_f \mid G\right)\right) P_\theta\left(G' \mid G, \neg s_f\right).$$

An invalid DAG G_i should have

$$P_\theta\left(G_i \mid G, \neg s_f\right).$$

1.2.2 Parametrization with Linear Transformer



$$Q = xW_Q \quad K = xW_K \quad V = xW_V$$

$$LinAttn_k(x) = \frac{\sum_{j=1}^{J} \left(\varphi(Q_k)^T \varphi(K_j)\right) V_j}{\left(\sum_{j=1}^{J} \left(\varphi(Q_k)^T \varphi(K_j)\right)\right.}$$

For any DAG G, we will define its reward as the joint probability,

$$R(G) = P(D, G) = P(G)P(D|G),$$

if the *detailed-balance conditions* are satisfied for all the states of. the GFlowNet, then this yields a sampling process with probability proportional to $R(G)$, that is, $p(G|D) \propto R(G)$.

1.3.1 Modularity & Computational Efficiency.

$$\log R(G) = \sum_{j=1}^{d} LocalScore\big(X_j \mid Pa_G(X_j)\big),$$

and if we add some edge $X_i \to X_j$ to form G', then we get

$$\log R(G') - \log R(G) =$$
$$LocalScore\big(X_j \mid Pa_G(X_j) \cup \{X_i\}\big) - LocalScore\big(X_j \mid Pa_G(X_j)\big).$$

The difference in local scores is called the *delta score* or the *incremental value*.

1.3.2 Off- Policy Learning

$$L(\theta) = \mathbb{E}_{\pi}\left[\left[\log \frac{R(G')P_B(G|G')P_\theta(s_f \mid G)}{R(G)P_\theta(G'|G)P_\theta(s_f|G')}\right]^2\right]$$

The Distribution $\pi(G \to G')$ can be

1. On-Policy: $P_\theta(G'|G)$ it self
2. Off-Policy: transitions $G \to G'$ are collected based on $P_\theta(G'|G)$, along with their corresponding delta score, and they are stored in a replay buffer. ε-greedy policy is also employed.

Results on the Real-world Dataset

Experiments on Sachs , **notes = 11, edges = 17, M = 874**. Results in the Paper:

Table 1: Learning protein signaling pathways from flow cytometry data (Sachs et al., 2005). All results include a 95% confidence interval estimated with bootstrap resampling.

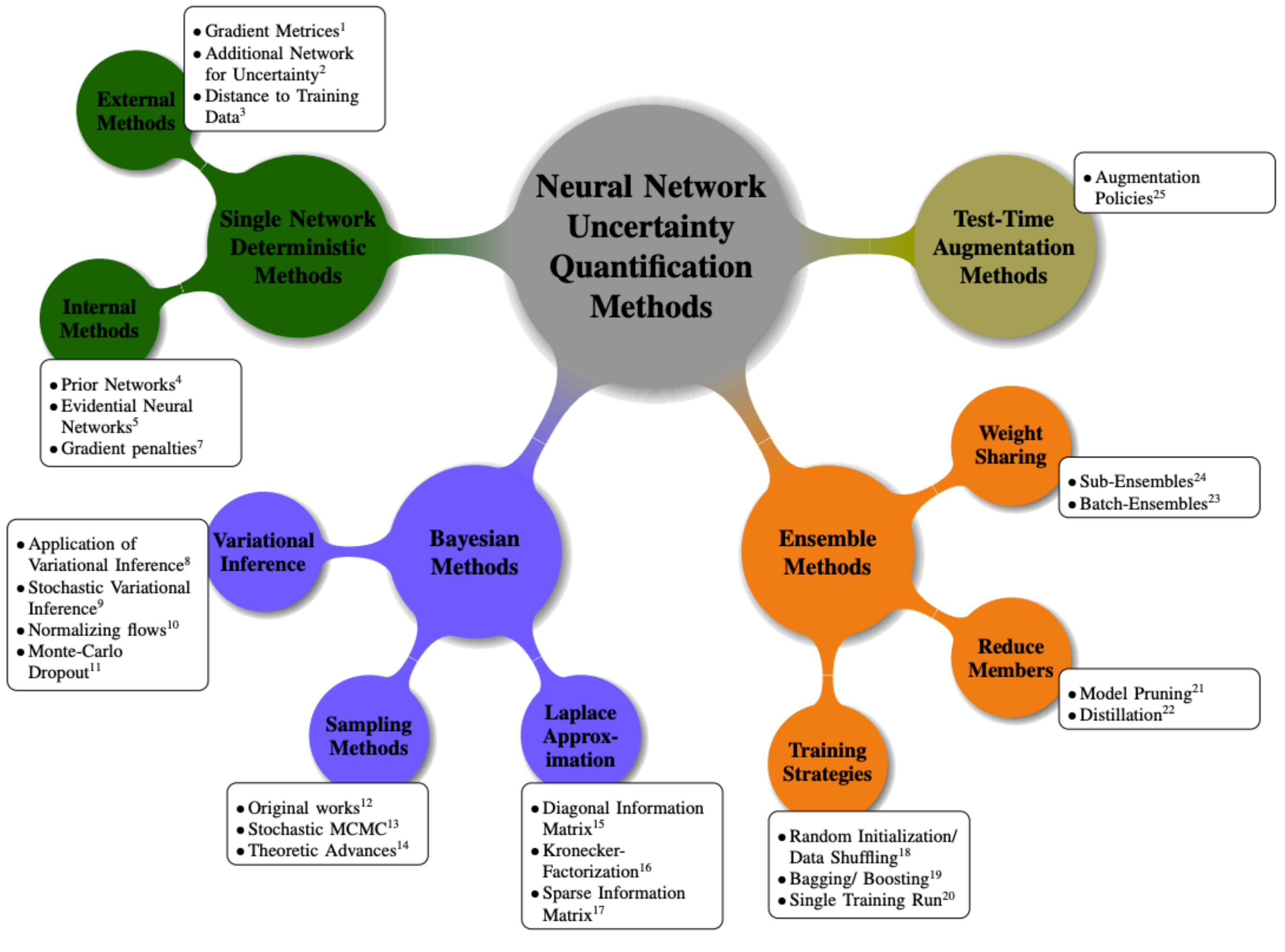
	$\mathbb{E}$-# Edges	$\mathbb{E}$-SHD	AUROC
MC^3	10.96 ± 0.09	22.66 ± 0.11	0.508
Gadget	10.59 ± 0.09	21.77 ± 0.10	0.479
Bootstrap GES	11.11 ± 0.09	23.07 ± 0.11	**0.548**
Bootstrap PC	7.83 ± 0.04	20.65 ± 0.06	0.520
DiBS	12.62 ± 0.16	23.32 ± 0.14	0.518
BCD Nets	4.14 ± 0.09	$\mathbf{18.14 \pm 0.09}$	0.510
DAG-GFlowNet	11.25 ± 0.09	22.88 ± 0.10	0.541

My results with different settings on Erdos-renyi prior

Prior	$\mathbb{E} - \#$ Edges	$\mathbb{E} - \text{SHD}$	AUROC
N=10 E/N=1	9.226	22.538	0.449
N=10 E/N=1.5	10.682	22.719	0.535

2. Uncertainty in Deep Learning (1)

The Four UQ Methods



Neural Network Uncertainty Quantification Methods
Single Network Deterministic Methods
External Methods
• Gradient Metrices[1] • Additional Network for Uncertainty[2] • Distance to Training Data[3]
Internal Methods
• Prior Networks[4] • Evidential Neural Networks[5] • Gradient penalties[7]
Test-Time Augmentation Methods
• Augmentation Policies[25]
Bayesian Methods
Variational Inference
• Application of Variational Inference[8] • Stochastic Variational Inference[9] • Normalizing flows[10] • Monte-Carlo Dropout[11]
Sampling Methods
• Original works[12] • Stochastic MCMC[13] • Theoretic Advances[14]
Laplace Approx-imation
• Diagonal Information Matrix[15] • Kronecker-Factorization[16] • Sparse Information Matrix[17]
Ensemble Methods
Weight Sharing
• Sub-Ensembles[24] • Batch-Ensembles[23]
Reduce Members
• Model Pruning[21] • Distillation[22]
Training Strategies
• Random Initialization/ Data Shuffling[18] • Bagging/ Boosting[19] • Single Training Run[20]

2.1.1 Bayesian Methods (VI, Sampling and LA)

1. Train such BNN efficiently/on large-scale tasks: SaprseVI, Variational Bayesian Last Layers, Symmetric-aware (these are all trying to explore the Weight Space)
2. Other ways: transformers can do bayesian inference
3. design the prior/sampling efficiency/sample in high-dim space. Efficiently LA (matrix factorizations)
4. For more complex posterior (sampling from, LA approx)
5. extend to other networks, applicability and extensibility

2.1.2 Ensemble Methods

1. The diversity: enable deep ensemble, Stochastic Ensemble
2. Fast Ensembling (train or inference): accelerate inference (Diffusion Schrödinger Bridge),distillation; sharing info/weights among members
3. Connected with other traditional CV tasks (like Semi/self-supervised learning)

2.1.3 Test Time Augmentation

Different Augmentations are performed on the input data at the time of testing, and then <u>multiple inferences</u> are made using these enhanced data <u>with single model</u> to estimate the uncertainty of the prediction through the distribution of these inferred results.

1. Uncertainty estimation: This method is particularly useful in cases where *the input is noisy or the sample distribution is skewed.*

2. Improve the robustness of the model: TTA method can improve the sensitivity of the model to small perturbations or noise in the input data by enhancing multiple generated samples for multiple reasoning when the model is not confident about a single input sample, thus enhancing the stability of the prediction.

- Pre-trained model friendly
- need to design the augmentation
- cost a lot

2.1.4 Single Networks

1. internal: directly considering the output as some distributions
2. external methods: calibrated by gradient information, external metrics
3. Dirichlet and Evidential Neural Networks

An Active Area: Conformal Prediction

2.2 An Active Area: Conformal Prediction

It is another way to do Uncertainty Quantification (predictive intervals).

An Introduction: https://github.com/pyladiesams/conformal-prediction-jan2024?tab=readme-ov-file

Some Young talents like

Ying Jin

Home Publications Talks Teaching Posts

Wojcicki-Troper Postdoc Fellow
@ HDSI & HMS

Contact
yjin at hcp.med.harvard.edu
180A Longwood Ave, Boston, MA

- Email
- Website
- Google scholar
- Curriculum Vitae
- GitHub

About me

Hi! I'm Ying. I am currently a Wojcicki-Troper Postdoctoral Fellow at Harvard Data Science Initiative, where I have the fortune to be mentored by Professor José Zubizarreta and also work with Professor Marinka Zitnik.

Starting July 2025, I will be an Assistant Professor in the Department of Statistics and Data Science at the Wharton School, University of Pennsylvania. I obtained my PhD in Statistics from Stanford University in 2024, advised by Professors Emmanuel Candès and Dominik Rothenhäusler. Here are my CV, Github and Google scholar pages.

I currently help organize the Online Causal Inference Seminar.

Research interests

I work on statistical problems related to two main themes:

- **Uncertainty quantification**
  I am interested in understanding the uncertainty of black-box AI models for their confident deployment in critical domains. Specifically, I think about what guarantees are meaningful, how to achieve them with easy-to-use, widely-applicable methods, and their applications in biomedical discovery and human decisions.

2.2 An Active Area: Conformal Prediction

Uncertainty Quantification over Graph with Conformalized Graph Neural Networks

Kexin Huang[1] **Ying Jin**[2] **Emmanuel Candès**[2,3] **Jure Leskovec**[1]

Some Young talents like

Heng Yang

Assistant Professor at Harvard SEAS
Robotics, Vision, Control, Optimization, Learning

I am an Assistant Professor of Electrical Engineering in the School of Engineering and Applied Sciences (SEAS) at Harvard University.

I direct the Computational Robotics Group at Harvard University. My group is broadly interested in the intersection of theory and practice, particularly computational algorithms that are robust, efficient, offer strong performance guarantees, and supercharge the next generation of intelligent systems.

I obtained my Ph.D. in Robotics from the Massachusetts Institute of Technology, where I was fortunate to work with Luca Carlone in the Laboratory for Information and Decision Systems.

I am also a part-time research scientist at NVIDIA Research.

Space Needle, 2022

The Benefit of Being Bayesian in Online Conformal Prediction

Zhiyu Zhang
Harvard University
zhiyuz@seas.harvard.edu

Zhou Lu
Princeton University
zhoul@princeton.edu

Heng Yang
Harvard University
hankyang@seas.harvard.edu

proposed on Arxiv on Oct 3, 2024

3. Makeup for Last Week

BayesDiff: Estimating Pixel-wise Uncertainty in Diffusion

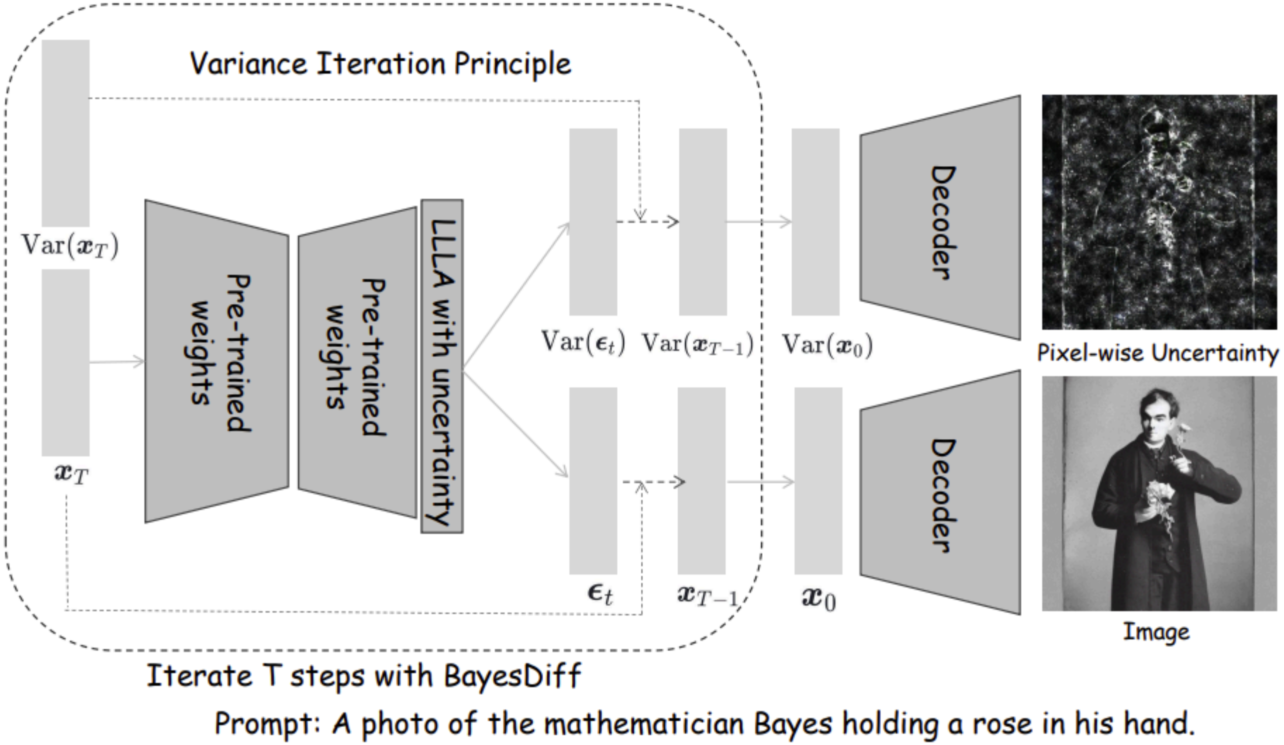


Figure 1: Given an initial point $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, our BayesDiff framework incorporates uncertainty into the denoising process and generates images with pixel-wise uncertainty estimates.

3.1 BayesDiff: Estimating Pixel-wise Uncertainty in Diffusion

3.1.1 The Meaning of Uncertainty

Figure 3: The images with the highest (left) and lowest (right) uncertainty among 5000 unconditional generations of U-ViT model trained on ImageNet at 256 × 256 resolution.

A photo of a bald eagle. A photo of an iguana. A photo of a white shark.

Figure 4: The images with the highest (left) and lowest (right) uncertainty among 80 generations on Stable Diffusion at 512 × 512 resolution.

3.1.2 Figure and Filter Our Low Quality Images

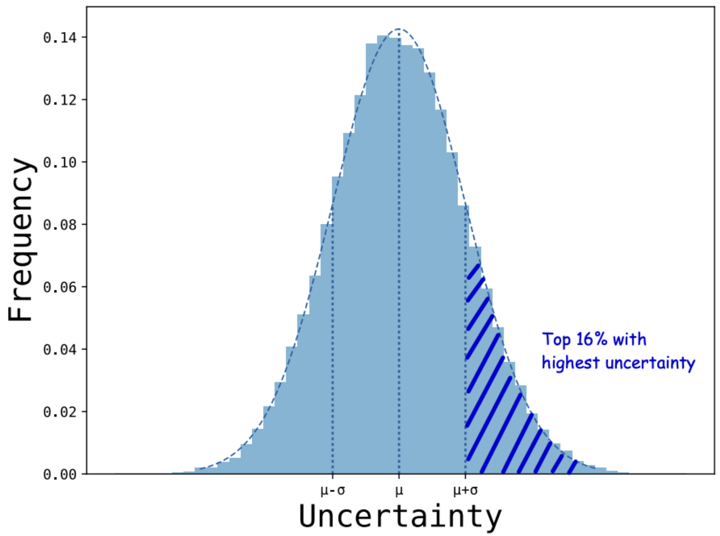


Figure 6: The empirical distribution of the uncertainty estimates yielded by our approach. The dashed line denotes the normal distribution fitted on them. Inspired by this, we propose to filter out the top 16% samples with the highest uncertainty.

3.1.3 Enhanced Diversity via BayesDiff

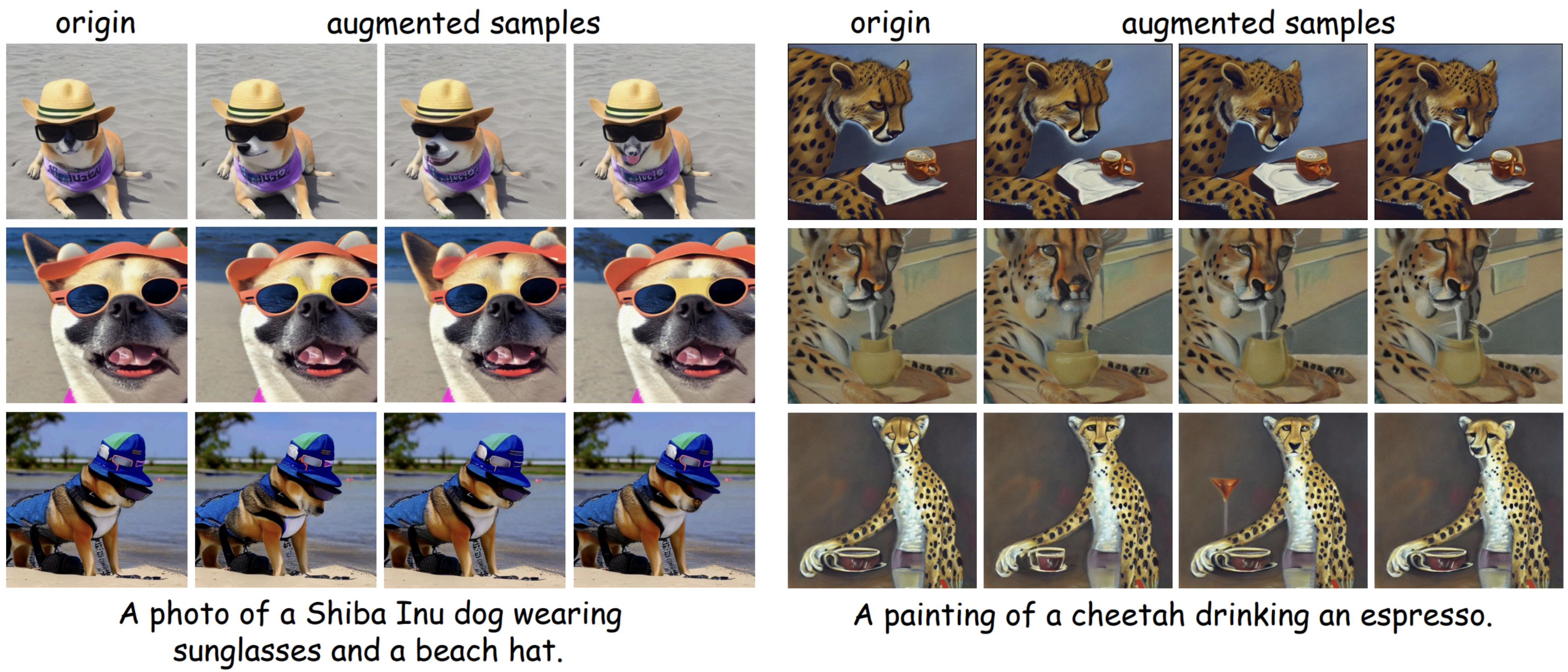
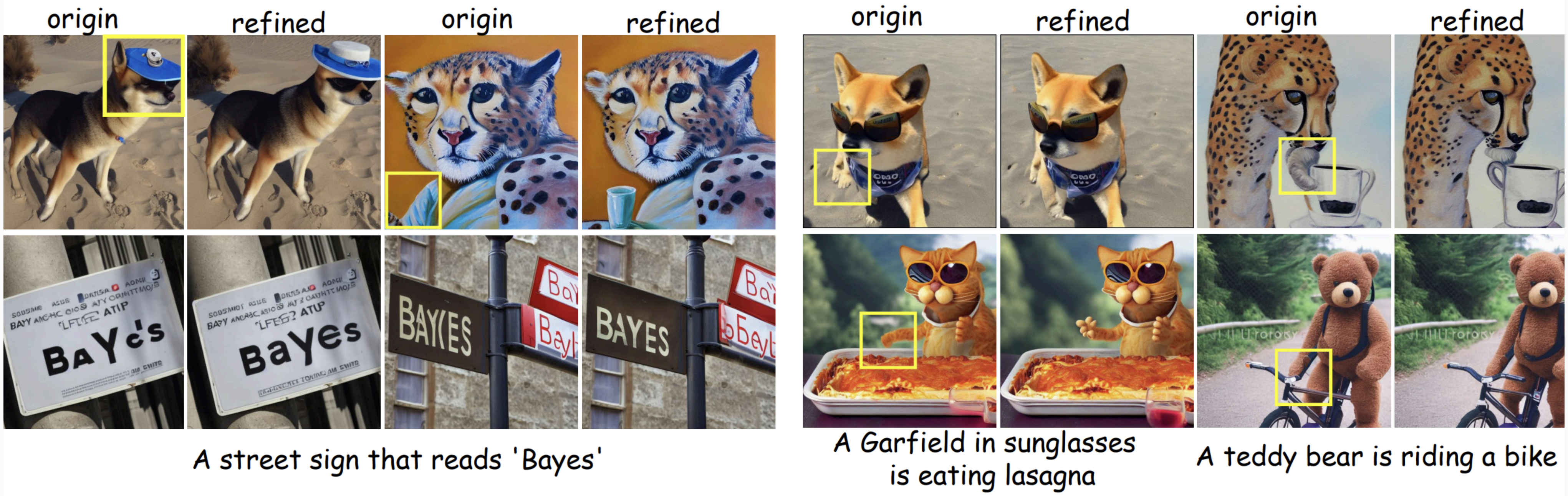


Figure 7: Examples of the augmentation of good generations with enhanced diversity on Stable Diffusion with DDIM sampler (50 NFE).

3.1 BayesDiff: Estimating Pixel-wise Uncertainty in Diffusion

3.1.4 Correction for the Failure Generations



ure 8: Examples of the rectification of artifacts and misalignment in failure generations on St
fusion with DDIM sampler (50 NFE). The flawed samples are identified by humans and
ınding boxes are manually annotated.

Experimental Results of My Naive Thoughts

Title: Uncertainty-guided Image Denoising: Experiments on CIFAR-10

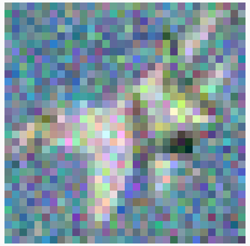
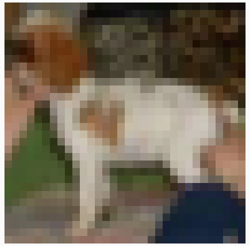
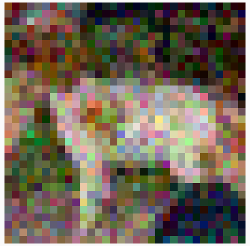
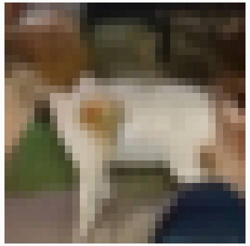
3.2.1 Basic Settings

- Clean Images: CIFAR-10 datasets
- Noisy Images: CIFAR-10 datasets with Gaussian Noise: $N(0, 0.1)$
- Base Methods: U-Net based Denoiser, MSE loss $= \frac{1}{N}\sum_{i}^{N}(y_i - f(x_i))^2$
- Uncertainty-Guided: loss $= \frac{1}{N}\sum_{i}^{N}(y_i - f(x_i))^2 + \alpha U_a$, U_a is the Aleatoric Uncertainty
- Baseline of U-Net Denoising on CIFAR-10: PSNR (Peak Signal to Noise Ratio) = 28.25 dB

3.2 Some Experimental Results of My Naive Thoughts

Methods	PSNR	Hyperparamters	UT/UE/UA
Basic U-Net	Avg: 28.21 db	\	.239/.088/.151
U-Guided	28.81 db	stepsize: 25, alpha: 1−>2 quadratic	**.237** /.086/.151
U-Guided	28.79 db	stepsize: 10, gamma = 0.1, alpha:1−>6 quadratic	.251/.091/.159
U-Guided	28.81 db	step_size: 5, gamma=0.3, alpha: 1−>6 Linear	**.238** /.087/.151
Clean Image	\	\	.101/.036/.065
Noisy Image	\	\	.678/.234/.445

3.2 Some Experimental Results of My Naive Thoughts

Clean	**Noisy**	**U-Net** (28.21dB)	**U-Guided** (28.81dB)
			
			

Looks similarly, but numerically succeed!

4. Plan for Next Week

4. Plan for Next Week

1. Continue to summarize the Uncertainty in Deep Learning Github Repository
2. Read some other Bayesian Structural Learning papers, try to visualize & compare the results
3. Extend reading on Uncertainty$\times$Generative Models
4. For the uncertainty-guided denoising attempts, try larger/more real-world datasets