

# Diffusion Prior for Online Decision Making

## A Case Study of Thompson Sampling

**Yu-Guan Hsieh**

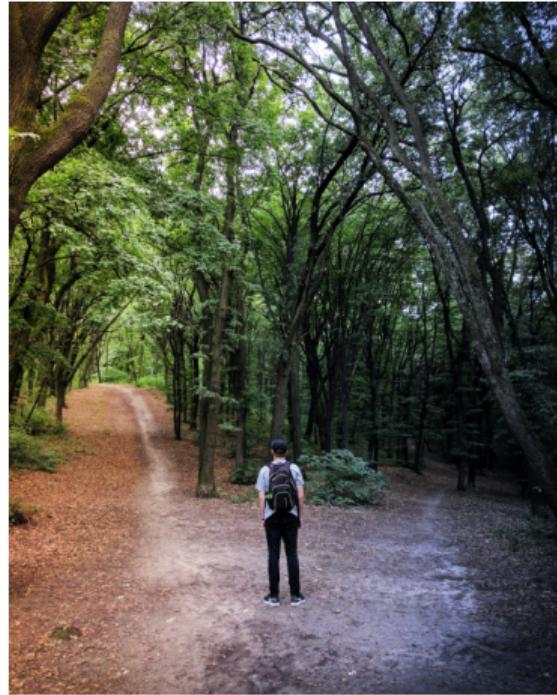
Supervisor: Shiva Kasiviswanathan

Mentor: Patrick Bloebaum

Also working with: Branislav Kveton

Internship from 08.01.2022 to 11.25.2022 in AWS causality team

# Uncertainty in Online Decision Making



# Prior Knowledge in Decision Making

san jose restaurants

**Yard House**  
4.3 ★★★★ (3,438) - \$  
New American - 300 Santana Row Suite 101  
Upscale sports bar with many draft beers  
Open - Closes 12AM  
The Bay's Best Margaritas

**Fogo de Chão Brazilian Steakhouse**  
4.3 ★★★★★ (2,549) - \$\$\$  
Brazilian - 377 Santana Row #1090  
Upmarket Brazilian churrascaria  
Closes soon - 10PM - Opens 11:30AM  
Tue  
Dine-in - Curbside pickup - No-contact delivery

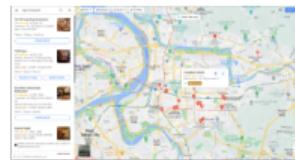
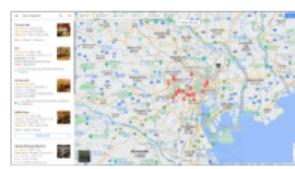
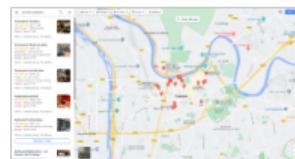
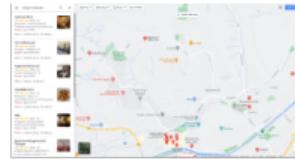
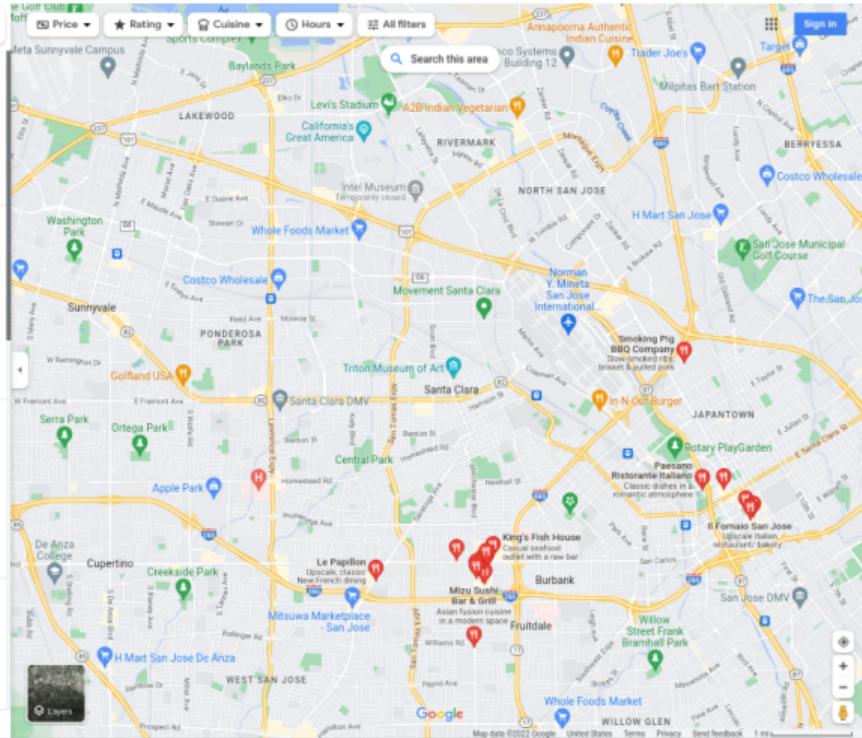
[RESERVE A TABLE](#) [ORDER ONLINE](#)

**Il Fornalo San Jose**  
4.2 ★★★★ (1,087) - \$\$  
Italian - 302 S Market St  
Upscale Italian restaurant/bakery chain  
Closes soon - 10PM - Opens 6:30AM  
Tue  
Dine-in - Curbside pickup - No-contact delivery

[RESERVE A TABLE](#) [ORDER ONLINE](#)

**Sauced BBQ & Spirits - Santana Row**  
4.3 ★★★★★ (309) - \$\$  
Barbecue - 3055 Olin Ave #1005  
Relaxed hangout with BBQ & drinks  
Closes soon - 10PM - Opens 11AM Tue  
Dine-in - Takeout - No-contact delivery

Update results when map moves



# Project Overview

Explore online decision making with prior described by deep generative model

# Project Overview

Explore online decision making with prior described by deep generative model

- Online decision making: **multi-armed bandits** with Thompson sampling

# Project Overview

Explore **online decision making** with **prior** described by **deep generative model**

- Online decision making: **multi-armed bandits** with Thompson sampling
- Deep generative prior: **denoising diffusion models**

# Project Overview

Explore **online decision making** with **prior** described by **deep generative model**

- Online decision making: **multi-armed bandits** with Thompson sampling
- Deep generative prior: **denoising diffusion models**
- Contributions
  - ▶ Design a Thompson sampling algorithm that runs with a given diffusion model
  - ▶ Design a training procedure to learn a diffusion model from **imperfect** data

# Project Overview

Explore **online decision making** with **prior** described by **deep generative model**

- Online decision making: **multi-armed bandits** with Thompson sampling
- Deep generative prior: **denoising diffusion models**
- Contributions
  - ▶ Design a Thompson sampling algorithm that runs with a given diffusion model
  - ▶ Design a training procedure to learn a diffusion model from **imperfect** data
- Benefit: a good prior grants better performance with limited data

# Plan

① Multi-Armed Bandits and Meta-Learning

② Denoising Diffusion / Score-Based Models

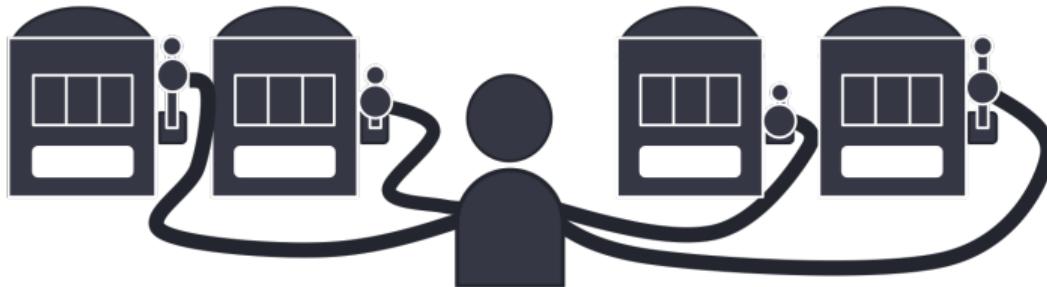
③ Algorithms

④ Numerical Experiments

⑤ Conclusion and Perspectives

# Multi-Armed Bandits

- Learner pulls arm  $a_t \in \mathcal{A} = \{1, \dots, K\}$  at round  $t$
- Learner receives rewards  $r_t$  drawn from the arm's distribution
- The goal is to maximize the cumulative rewards  $\sum_t r_t$
- Applications: recommendation systems, online advertisement, clinical trial, ...



# Thompson Sampling

- A Bayesian approach to tackle multi-armed bandits
- The decision is random
- Has often better empirical performance than UCB (frequentist and deterministic)

# Thompson Sampling

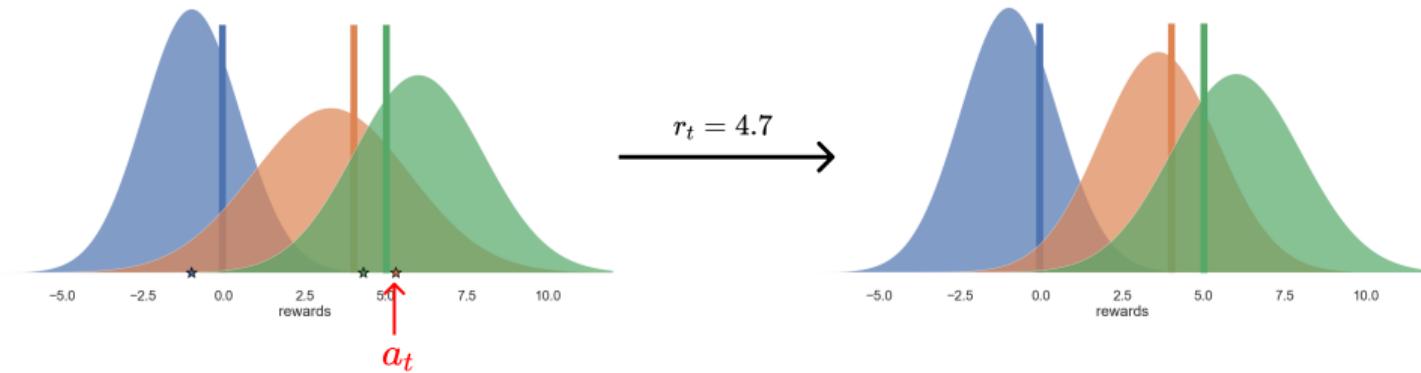
- A Bayesian approach to tackle multi-armed bandits
- The decision is random
- Has often better empirical performance than UCB (frequentist and deterministic)
- Precisely, for the parameter of interest  $w$  it maintains posterior distribution

$$p(w | \mathcal{H}) \propto p(\mathcal{H} | w)p(w)$$

where  $p(w)$  is a prior over  $w$  and  $\mathcal{H} = (a_s, r_s)_{s \in \{1, \dots, t\}}$  is the interaction history

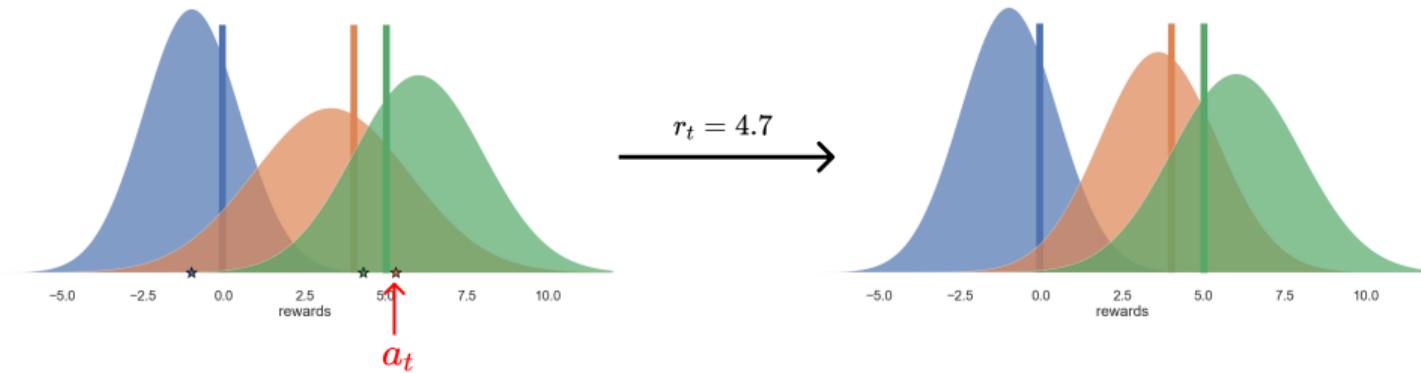
# Thompson Sampling

- In vanilla MAB with known noise distribution, the parameter of interest is the vector of expected reward  $\mu = (\mu^a)_{a \in \mathcal{A}}$
- At each round, we sample  $\tilde{\mu}$  from the posterior distribution  $\mathbb{P}(\mu | \mathcal{H})$  and pull the arm with the highest mean  $a \in \arg \max_{a \in \mathcal{A}} \tilde{\mu}^a$



# Thompson Sampling

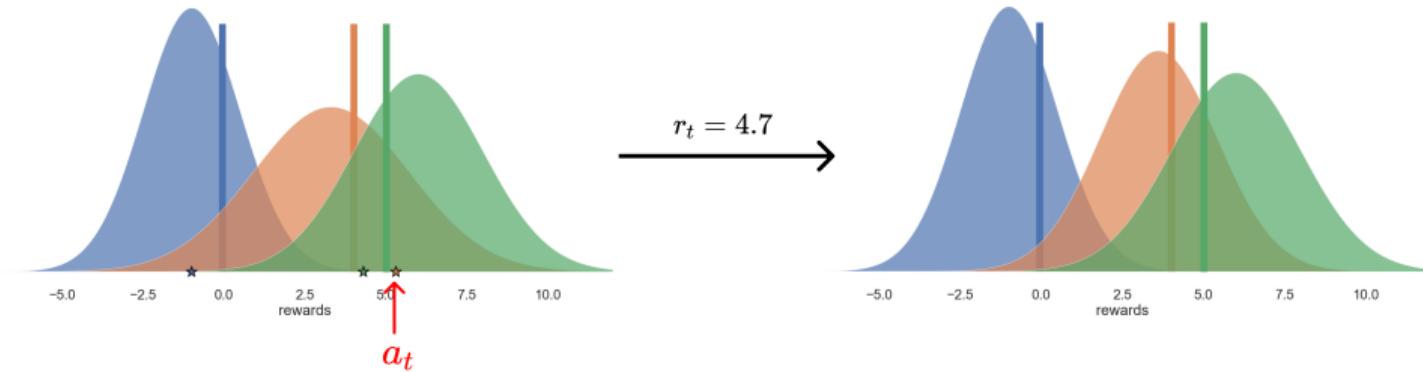
- The algorithm is sensitive to the choice of prior



# Thompson Sampling

- The algorithm is sensitive to the choice of prior

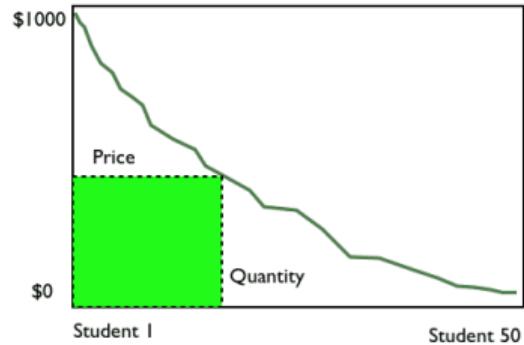
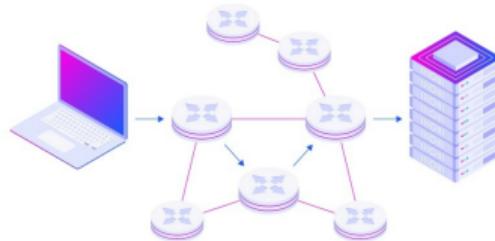
*Can we learn the prior?*



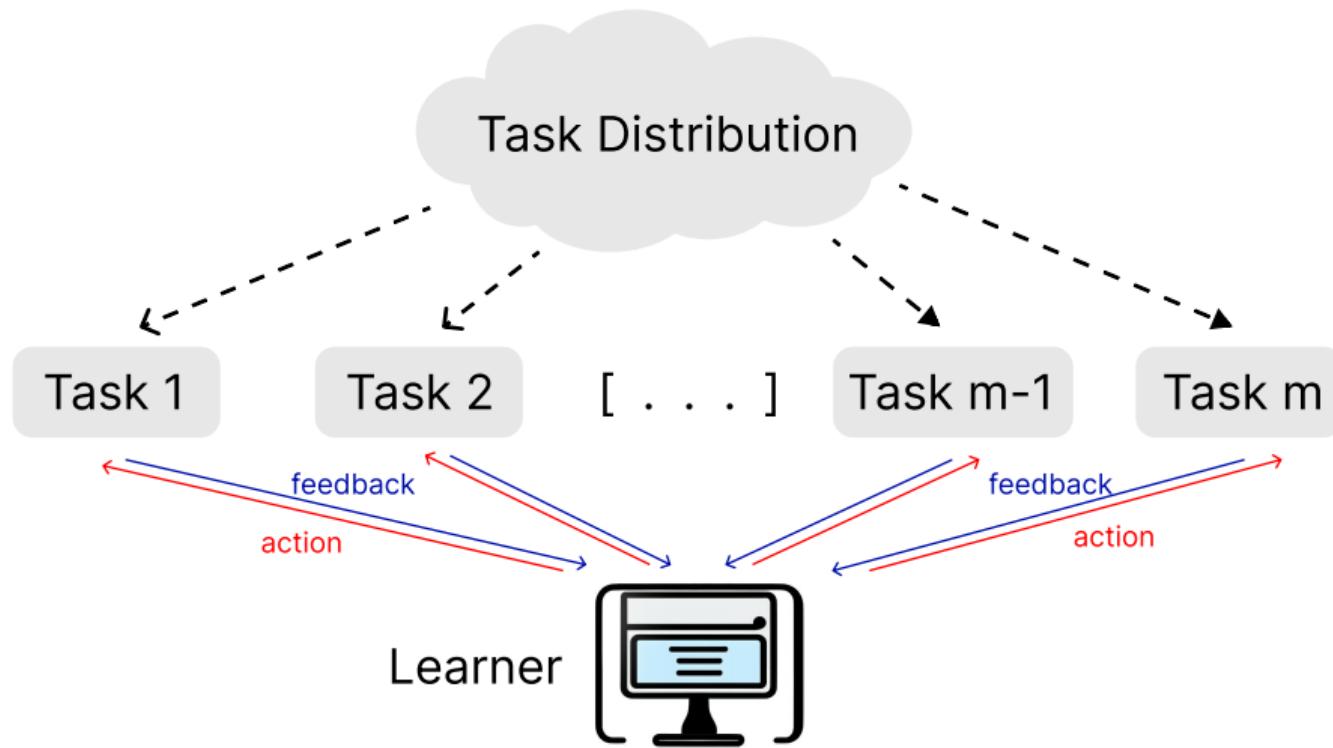
# A Class of Bandit Tasks

- Recommend items to different customers
- Solve online shortest routing in different networks
- Assign price to different items using an online pricing algorithm

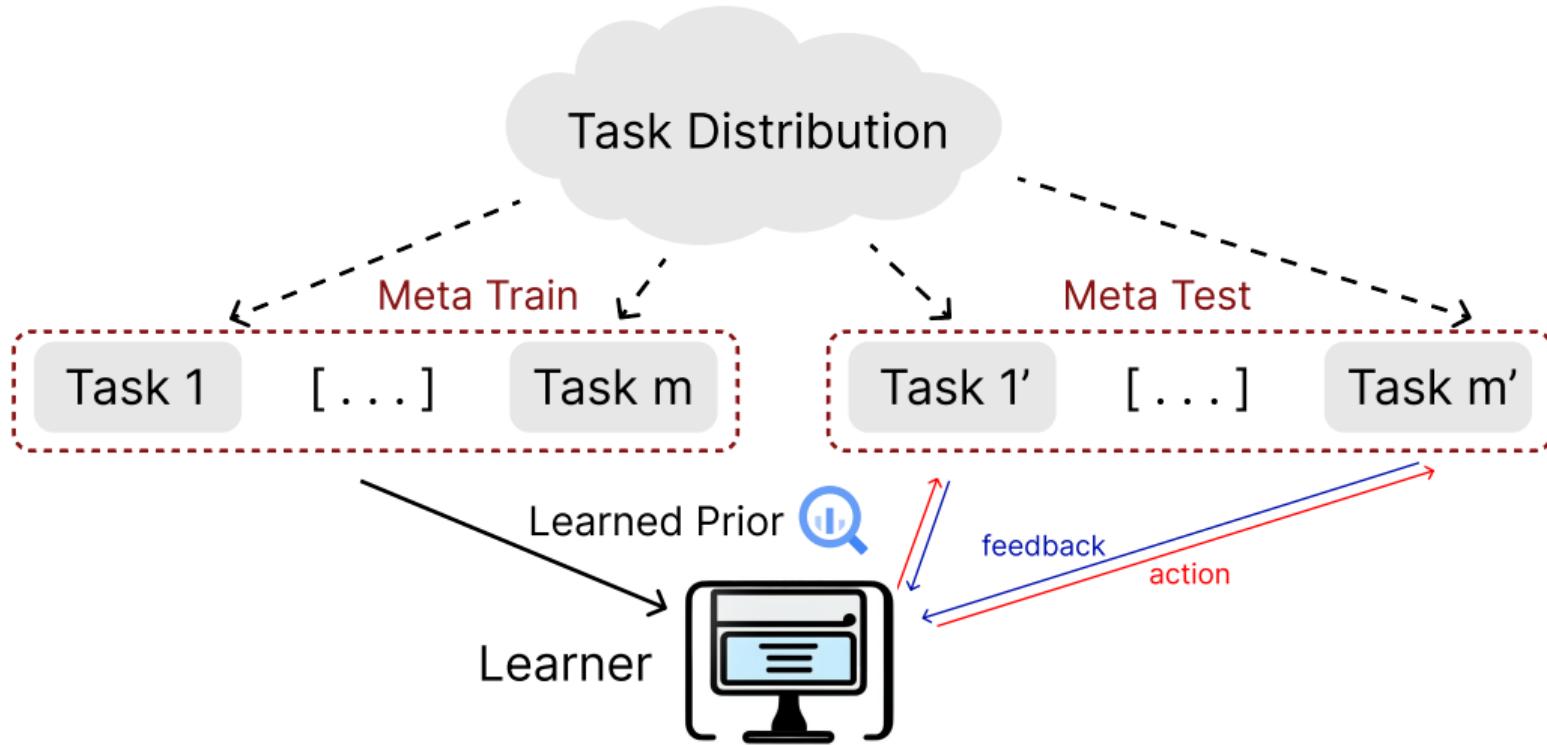
	1	2	3	4	5	6
a	+	?	-	-	?	-
b	-		+			+
c	+	+	-	-	-	-
d		+	+	+	-	
e	-	-		+	+	



# Meta Learning a Prior for Bandits



# Meta Learning a Prior for Bandits



# Plan

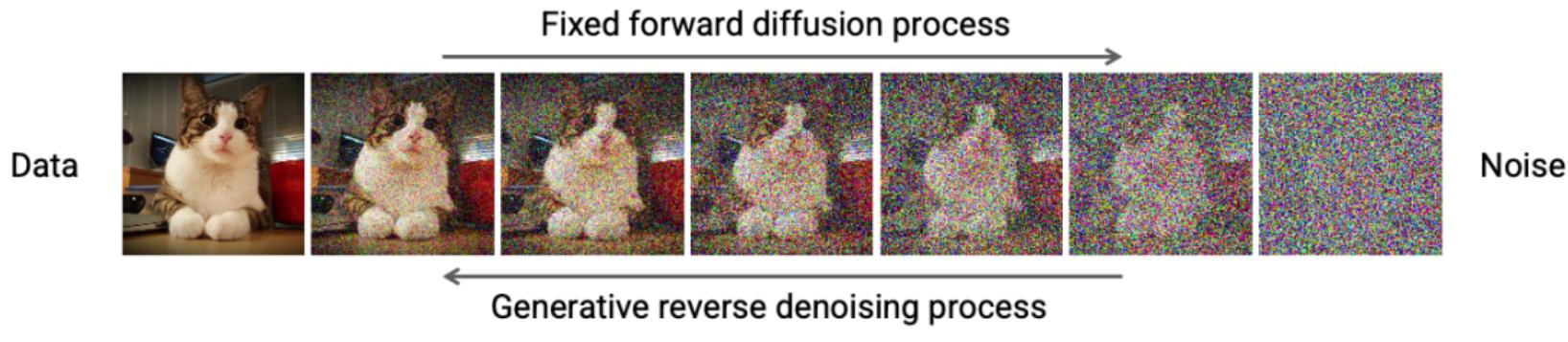
- ① Multi-Armed Bandits and Meta-Learning
- ② Denoising Diffusion / Score-Based Models
- ③ Algorithms
- ④ Numerical Experiments
- ⑤ Conclusion and Perspectives

# The Rise of Diffusion Models

- State of the art image generation models: Imagen, Dalle-2, Midjourney, Stable Diffusion
- And beyond: audio synthesis, molecular generation, RL trajectories



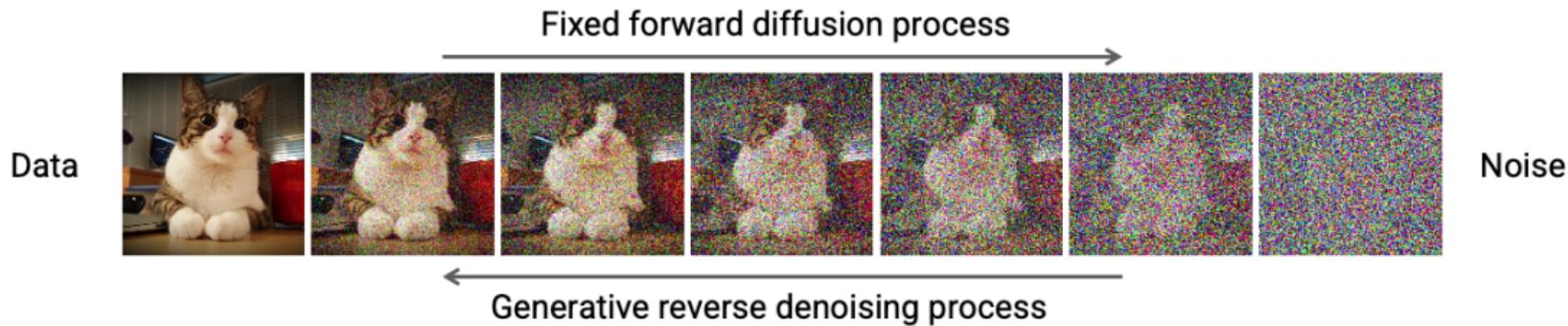
# Diffusion Models in a Nutshell



(Source: 2022 CVPR diffusion model tutorial)

- Add noise in the forward process:  $q(X_{\ell+1} | x_\ell) = \mathcal{N}(X_{\ell+1}; \sqrt{\alpha_{\ell+1}}x_\ell, (1 - \alpha_{\ell+1})I)$
  - Parameterize the reverse process with a denoiser  $h_\theta$       both are Gaussian by construction
- $$p_\theta(X_\ell | x_{\ell+1}) = q(X_\ell | x_{\ell+1}, X_0 = h_\theta(x_{\ell+1}, \ell+1)) \propto \overbrace{q(x_{\ell+1} | X_\ell)q(X_\ell | X_0 = h_\theta(x_{\ell+1}, \ell+1))}^{\text{both are Gaussian by construction}}$$

# Diffusion Models in a Nutshell



(Source: 2022 CVPR diffusion model tutorial)

- The denoiser is trained to ‘denoise’
- Diffusion model as maximum likelihood estimation / reverse-time SDE
- **The iterative sampling process allows for better posterior sampling**

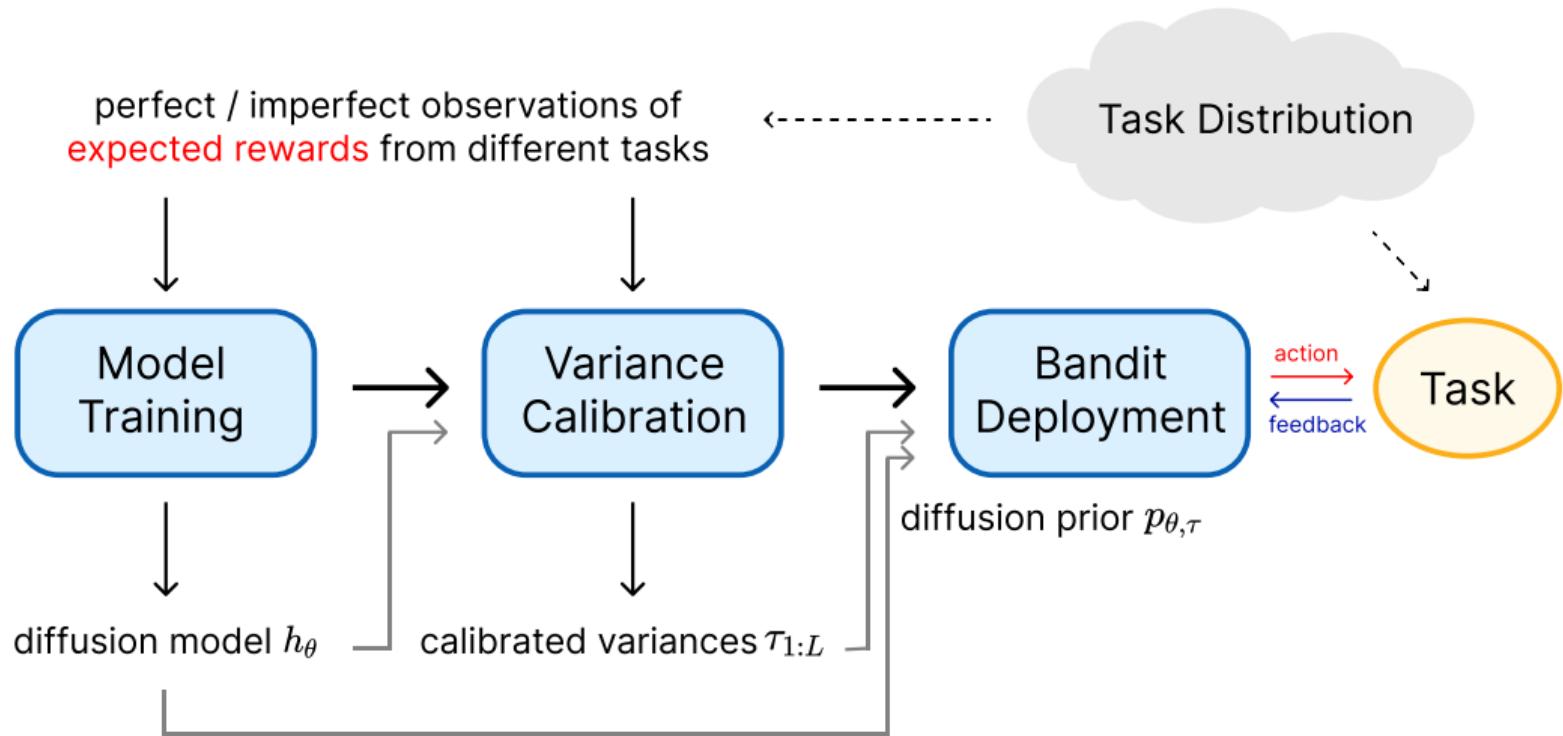
# Gaussian Prior versus Diffusion Prior

	Gaussian Prior	Diffusion Prior
Model Learning	Maximum likelihood Closed-form, fast	Deep learning Harder and slower
Posterior sampling	Closed-form, fast	Approximate, slower
Expressive power	Limited	Strong
Data efficiency	Bad?	Good?

# Plan

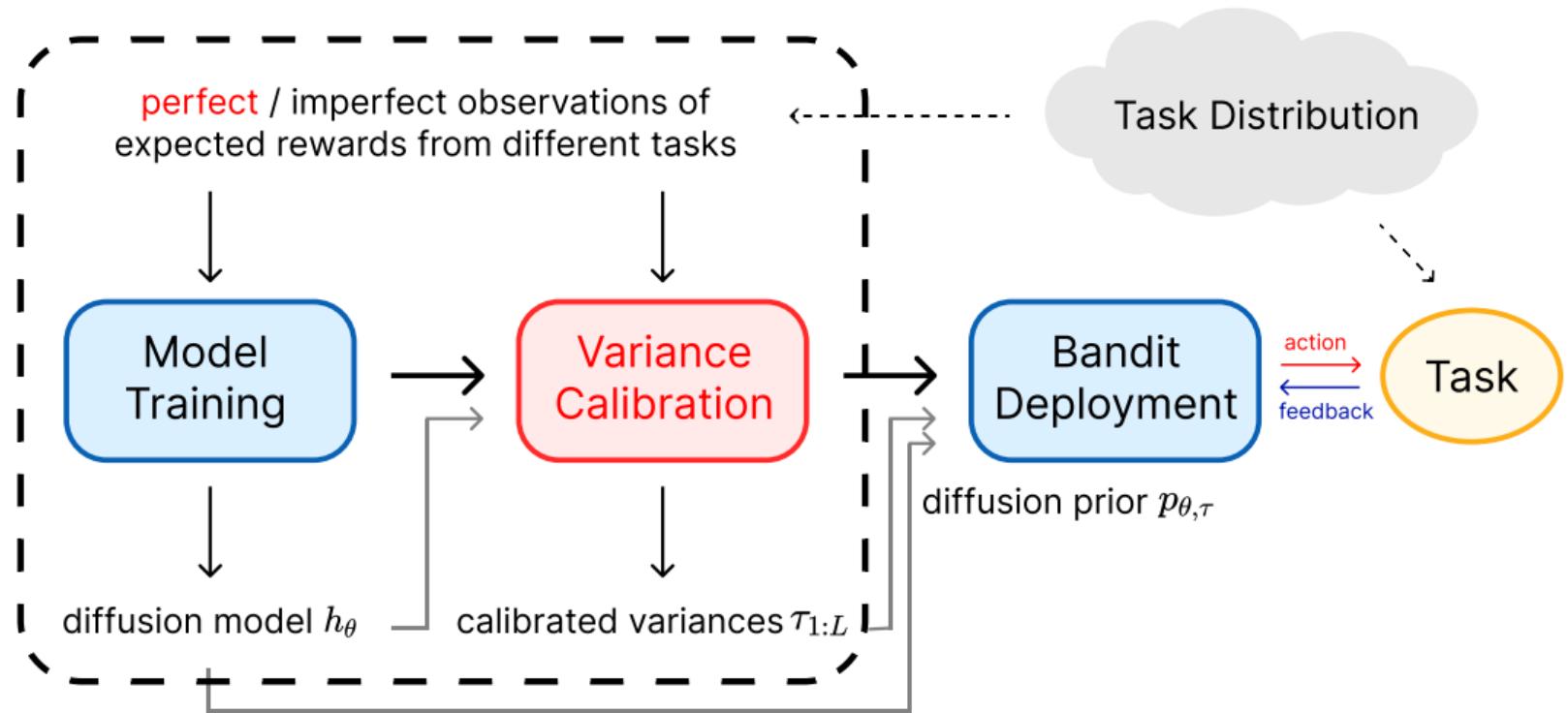
- ① Multi-Armed Bandits and Meta-Learning
- ② Denoising Diffusion / Score-Based Models
- ③ Algorithms
- ④ Numerical Experiments
- ⑤ Conclusion and Perspectives

# Overview



Assume that a trained diffusion model is provided

# Variance Calibration



# Variance Calibration

Goal: Calibrate the variance of the reverse diffusion process  $p_\theta(X_\ell | x_{\ell+1})$

- The variance of the original  $p_\theta(X_\ell | x_{\ell+1})$  is suboptimal: overly confident

# Variance Calibration

Goal: Calibrate the variance of the reverse diffusion process  $p_\theta(X_\ell | x_{\ell+1})$

- The variance of the original  $p_\theta(X_\ell | x_{\ell+1})$  is suboptimal: overly confident
- Instead, consider

$$p_{\theta,\tau}(X_\ell | x_{\ell+1}) = \int q(X_\ell | x_{\ell+1}, x_0) p'_{\theta,\tau}(x_0 | x_{\ell+1}) dx_0$$

where

- ▶  $p'_{\theta,\tau}(X_0 | x_{\ell+1})$  is a Gaussian distribution centered at  $\hat{x}_0 = h_\theta(x_{\ell+1}, \ell + 1)$  with covariance  $\text{diag}(\tau_{\ell+1}^2)$
- ▶  $\tau^2$  is the **mean squared reconstruction error**  $\tau_\ell^a = \sqrt{\mathbb{E}_{X_0, X_\ell} [\|X_0^a - h_\theta^a(X_\ell, \ell)\|^2]}$

# Variance Calibration

Goal: Calibrate the variance of the reverse diffusion process  $p_\theta(X_\ell | x_{\ell+1})$

$\tau^2$  is the mean squared reconstruction error  $\tau_\ell^a = \sqrt{\mathbb{E}_{X_0, X_\ell} [\|X_0^a - h_\theta^a(X_\ell, \ell)\|^2]}$

- $\tau^2$  can be easily estimated when having access to the **exact expected rewards**  $x_0 = \mu$  from different tasks

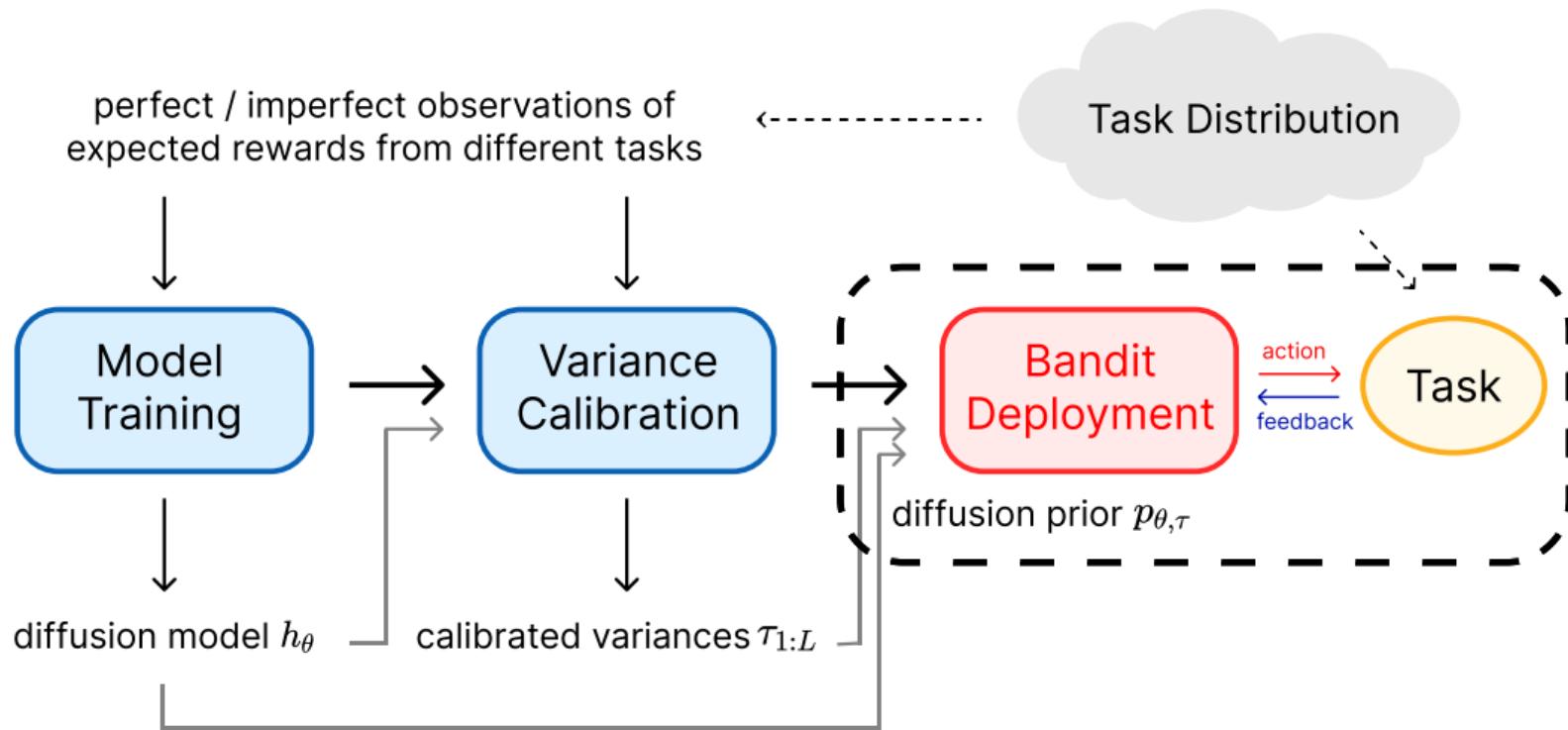
# Variance Calibration

Goal: Calibrate the variance of the reverse diffusion process  $p_\theta(X_\ell | x_{\ell+1})$

$\tau^2$  is the mean squared reconstruction error  $\tau_\ell^a = \sqrt{\mathbb{E}_{X_0, X_\ell} [\|X_0^a - h_\theta^a(X_\ell, \ell)\|^2]}$

- $\tau^2$  can be easily estimated when having access to the **exact expected rewards**  $x_0 = \mu$  from different tasks
- We also develop method to estimate  $\tau^2$  from incomplete and noisy data

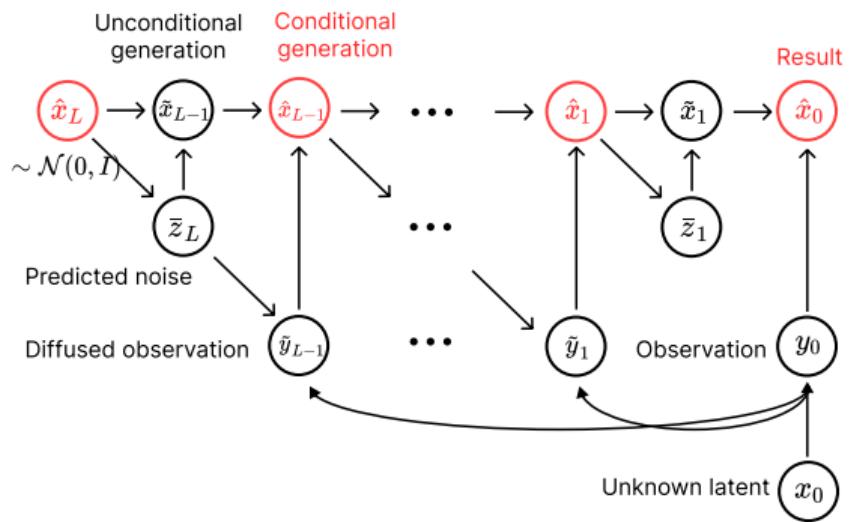
# Thompson Sampling with Diffusion Prior



# Thompson Sampling with Diffusion Prior

Goal: Sample from  $p_{\theta, \tau}(X_0 | y_0)$  provided imperfect observation  $y_0$

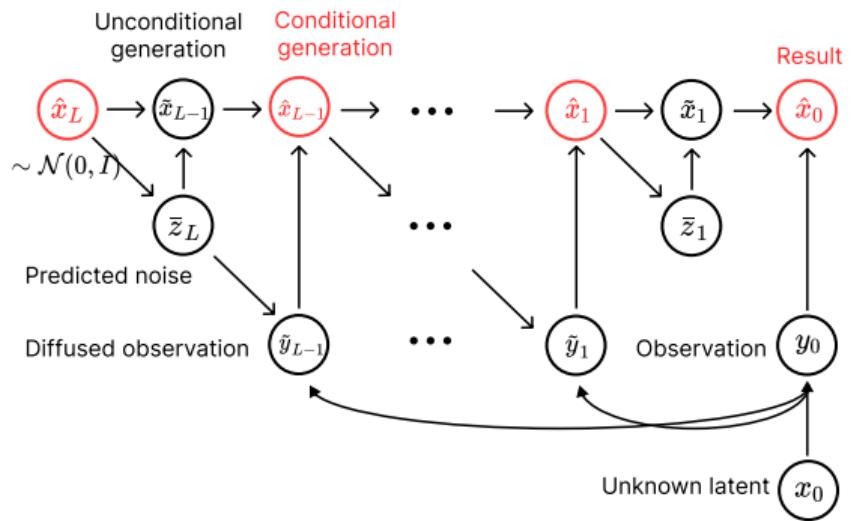
- In MAB,  $y_0 = \mathcal{H}$  is the history



# Thompson Sampling with Diffusion Prior

Goal: Sample from  $p_{\theta, \tau}(X_0 | y_0)$  provided imperfect observation  $y_0$

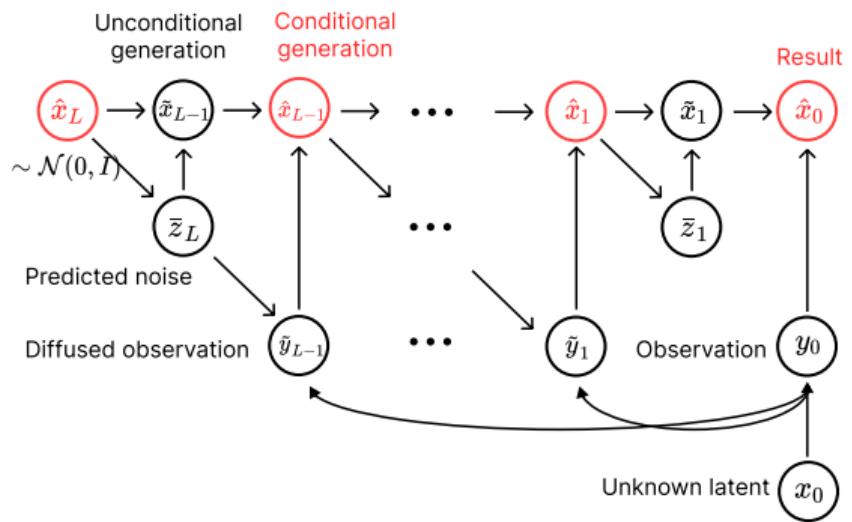
- In MAB,  $y_0 = \mathcal{H}$  is the history
- Condition the reverse process on  $y_0$ 
  - Sample  $x_L$  from  $X_L | y_0$
  - Sample  $x_\ell$  from  $X_\ell | x_{\ell+1}, y_0$



# Thompson Sampling with Diffusion Prior

Goal: Sample from  $p_{\theta,\tau}(X_0 | y_0)$  provided imperfect observation  $y_0$

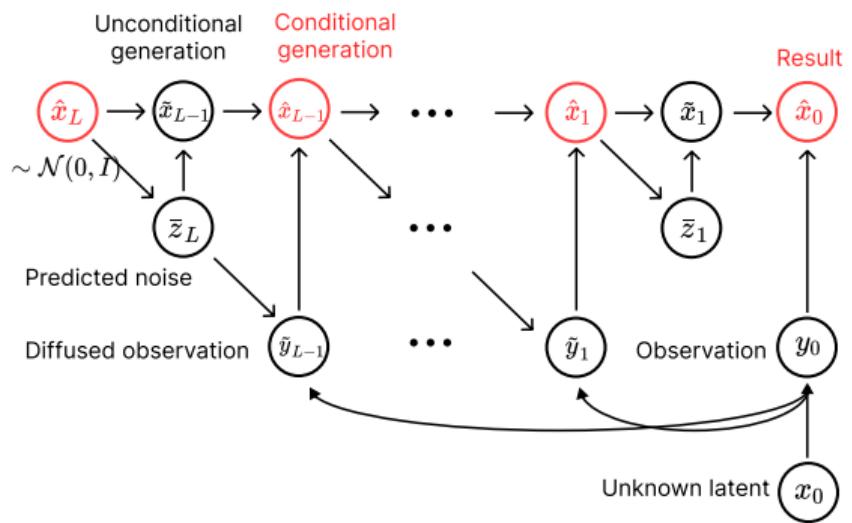
- In MAB,  $y_0 = \mathcal{H}$  is the history
- Condition the reverse process on  $y_0$ 
  - Sample  $x_L$  from  $X_L | y_0$
  - Sample  $x_\ell$  from  $X_\ell | x_{\ell+1}, y_0$
- Initialization: Sampled from  $\mathcal{N}(0, I)$



# Thompson Sampling with Diffusion Prior

Goal: Sample from  $p_{\theta,\tau}(X_0 | y_0)$  provided imperfect observation  $y_0$

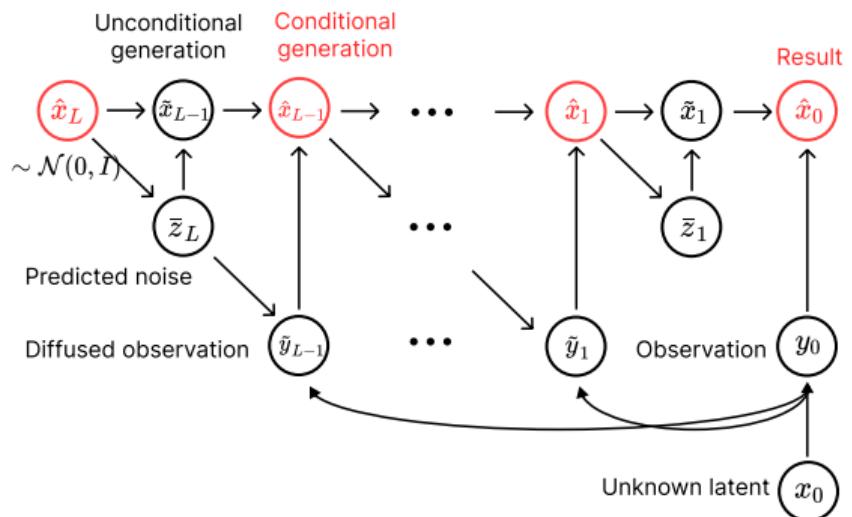
- In MAB,  $y_0 = \mathcal{H}$  is the history
- Condition the reverse process on  $y_0$ 
  - Sample  $x_L$  from  $X_L | y_0$
  - Sample  $x_\ell$  from  $X_\ell | x_{\ell+1}, y_0$
- Initialization: Sampled from  $\mathcal{N}(0, I)$
- Recursion: Mix an unconditional sampled  $\tilde{x}_\ell$  with a *diffused*  $\tilde{y}_\ell$



# Thompson Sampling with Diffusion Prior

Goal: Sample from  $p_{\theta, \tau}(X_0 | y_0)$  provided imperfect observation  $y_0$

- In MAB,  $y_0 = \mathcal{H}$  is the history
- Condition the reverse process on  $y_0$ 
  - Sample  $x_L$  from  $X_L | y_0$
  - Sample  $x_\ell$  from  $X_\ell | x_{\ell+1}, y_0$
- Initialization: Sampled from  $\mathcal{N}(0, I)$
- Recursion: Mix an unconditional sampled  $\tilde{x}_\ell$  with a *diffused*  $\tilde{y}_\ell \rightarrow$  How?



# Thompson Sampling with Diffusion Prior

Goal: Sample from  $p_{\theta,\tau}(X_0 | y_0)$  provided imperfect observation  $y_0$

- For arm  $a$  that has never been pulled, set  $\tilde{q}(x_\ell^a | x_{\ell+1}, y_0) = p_{\theta,\tau}(x_\ell^a | x_{\ell+1})$

# Thompson Sampling with Diffusion Prior

Goal: Sample from  $p_{\theta,\tau}(X_0 | y_0)$  provided imperfect observation  $y_0$

- For arm  $a$  that has never been pulled, set  $\tilde{q}(x_\ell^a | x_{\ell+1}, y_0) = p_{\theta,\tau}(x_\ell^a | x_{\ell+1})$
- For arm  $a$  that has been pulled at least once
  - ▶  $\hat{\mu}_t^a$  empirical mean;  $\sigma_t^a$  scaled noise standard deviation
  - ▶  $\bar{z}_{\ell+1}$  noise predicted by the denoiser from  $x_{\ell+1}$
  - ▶  $\tilde{y}_\ell^a = \sqrt{\bar{\alpha}_\ell} \hat{\mu}_t^a + \sqrt{1 - \bar{\alpha}_\ell} \bar{z}_{\ell+1}^a$  the diffused observation [where  $\bar{\alpha}_\ell = \prod_{k=1}^{\ell} \alpha_k$ ]

Set  $\tilde{q}(x_\ell^a | x_{\ell+1}, y_0) \propto \underbrace{p_{\theta,\tau}(x_\ell^a | x_{\ell+1})}_{\text{prior}} \mathcal{N}\left(x_\ell^a; \underbrace{\tilde{y}_\ell^a}_{\text{observation}}, \underbrace{\bar{\alpha}_\ell((\sigma_t^a)^2 + \rho_\ell(\tau_{\ell+1}^a)^2)}_{\text{denoising variance}}\right)$

**Algorithm** Thompson Sampling with Diffusion Prior (DiffTS)

1: **Input:** Trained denoiser  $h_\theta$ , denoising variance  $(\tau_\ell^2)_{\ell \in \{1, \dots, L\}}$ , presumed noise std  $\sigma'$

2: **for**  $t = 1, \dots$  **do** Posterior Sampling

3:   Sample  $x_L \sim \mathcal{N}(0, I)$

4:   **for**  $\ell \in L - 1, \dots, 0$  **do**

5:     Predict clean sample  $\hat{x}_0 = h_\theta(x_{\ell+1}, \ell + 1)$  and associated noise  $\bar{z}_{\ell+1}$

6:     Compute diffused observation  $\tilde{y}_\ell^a = \sqrt{\bar{\alpha}_\ell} \hat{\mu}_{t-1}^a + \sqrt{1 - \bar{\alpha}_\ell} \bar{z}_{\ell+1}$

7:     **for**  $a \in \mathcal{A}$  **do**

8:       If  $N_{t-1}^a = 0$ , sample  $x_\ell^a \sim p_{\theta, \tau}(X_\ell^a | x_{\ell+1})$

9:       If  $N_{t-1}^a > 0$ , sample

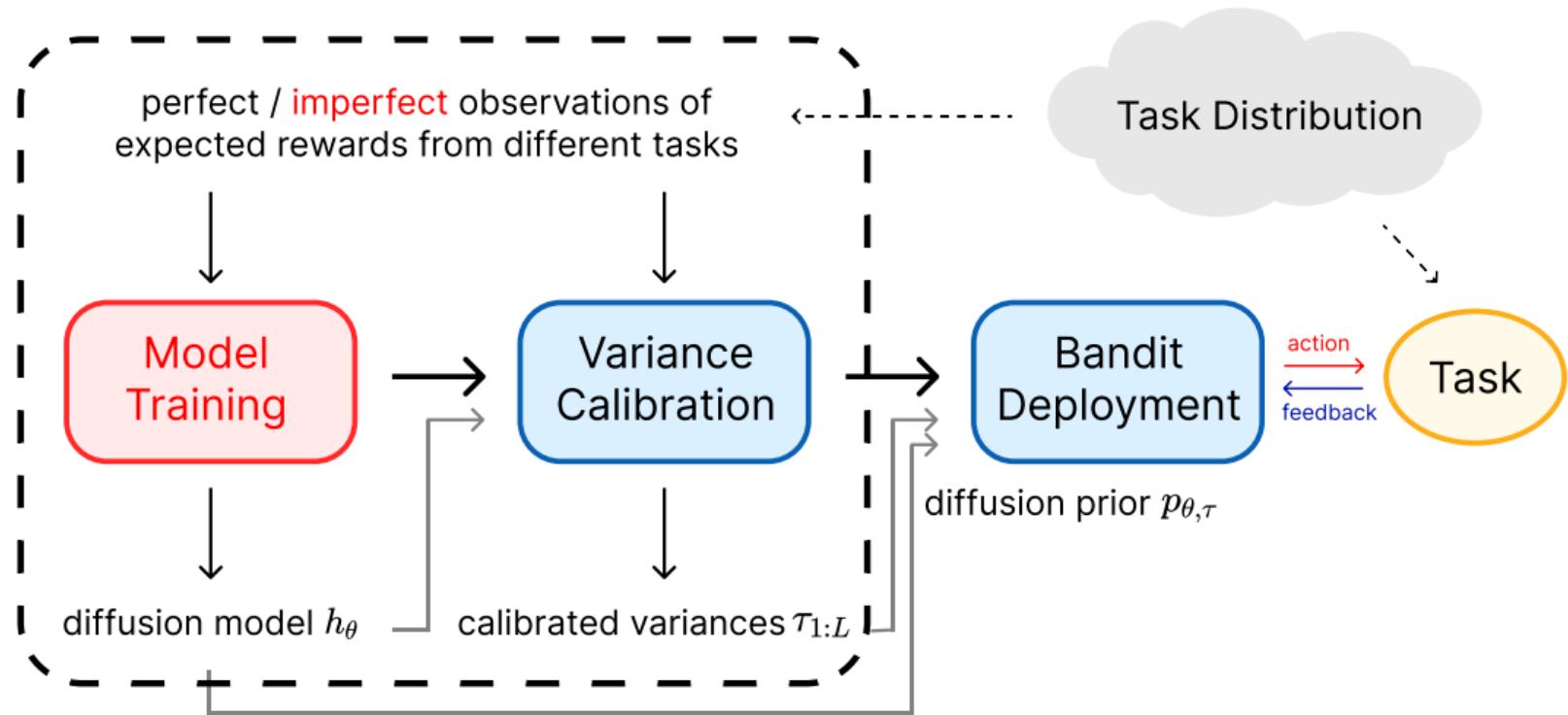
$$x_\ell^a \sim \tilde{q}(X_\ell^a | x_{\ell+1}, y_0) \propto p_{\theta, \tau}(X_\ell^a | x_{\ell+1}) \mathcal{N}(X_\ell^a; \tilde{y}_\ell^a, \bar{\alpha}_\ell((\sigma_t^a)^2 + \rho_\ell(\tau_{\ell+1}^a)^2))$$

10:      Pull arm  $a_t \in \arg \max_{a \in \mathcal{A}} x_0^a$

11:      Update number of pulls  $N_t^a$ , scaled std  $\sigma_t^a$ , and empirical reward  $\hat{\mu}_t^a$  for  $a \in \mathcal{A}$

Back to the training of diffusion model

# Model Training



# Model Training

Goal: minimize mean squared loss  $\mathbb{E}_{\ell, X_0, X_\ell} [\|X_0 - h_\theta(X_\ell, \ell)\|^2]$

- Training from perfect data  $x_0$ : minimize standard diffusion loss

$$\mathbb{E}_{\ell, x_0, x_\ell \sim X_\ell | x_0} [\|x_0 - h_\theta(x_\ell, \ell)\|^2]$$

# Model Training

Goal: minimize mean squared loss  $\mathbb{E}_{\ell, X_0, X_\ell} [\|X_0 - h_\theta(X_\ell, \ell)\|^2]$

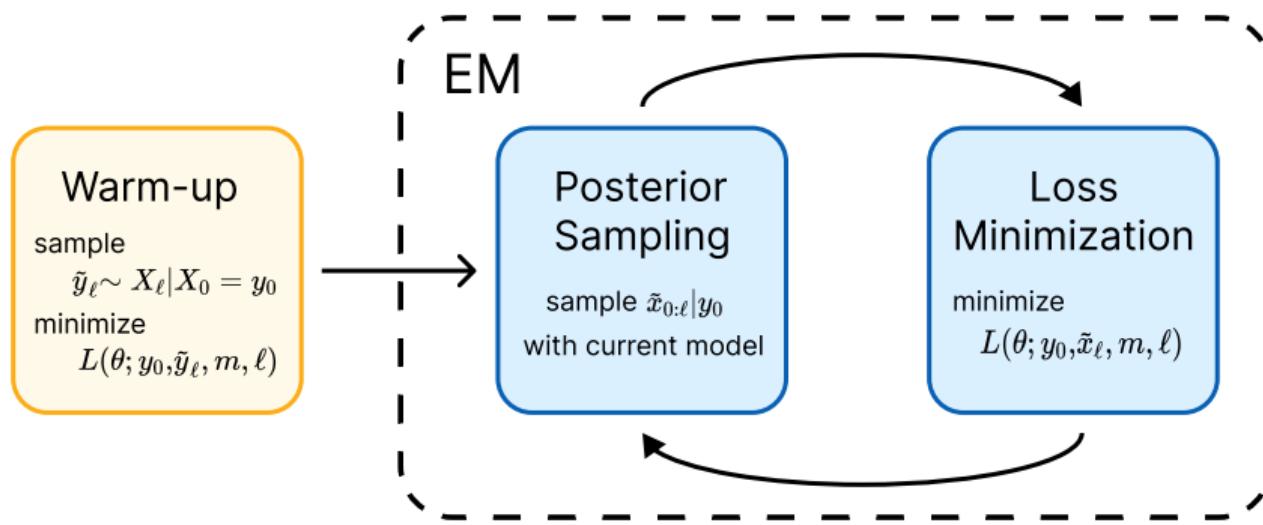
- Training from perfect data  $x_0$ : minimize standard diffusion loss

$$\mathbb{E}_{\ell, x_0, x_\ell \sim X_\ell | x_0} [\|x_0 - h_\theta(x_\ell, \ell)\|^2]$$

- Contribution: Training from **incomplete** and **noisy** data  $y_0 = m \odot (x_0 + z)$  where
  - ▶  $m \in \{0, 1\}^K$  is a binary mask
  - ▶  $z$  is a noise vector sampled from  $\mathcal{N}(0, \sigma^2 I)$

Challenge: both  $x_0$  and  $x_\ell$  are not available

# Training from Incomplete and Noisy Data



$$L(\theta; y_0, \tilde{x}_\ell, m, \ell) = \|m \odot y_0 - m \odot h_\theta(\tilde{x}_\ell, \ell)\|^2 + 2\lambda\sqrt{\bar{\alpha}_\ell}\sigma^2 \mathbb{E}_{b \sim \mathcal{N}(0, I)} b^\top \left( \frac{h_\theta(\tilde{x}_\ell + \varepsilon b, \ell) - h_\theta(\tilde{x}_\ell, \ell)}{\varepsilon} \right)$$

ignored masked value      MC-SURE regularization to counter noise

# Training from Incomplete and Noisy Data: Working Examples

5 0 4 1 9 2 1 3 1  
4 3 5 3 6 1 7 2 8  
6 9 4 0 9 1 1 2 4  
3 2 7 3 8 6 9 0 5  
6 0 7 6 1 8 7 9 3  
9 8 5 9 3 3 0 7 4  
9 8 0 9 4 1 4 4 6  
0 4 5 6 1 0 0 1 7  
1 6 3 0 2 1 1 7 9

Clean samples

5 0 4 1 9 2 1 3 1  
4 3 5 3 6 1 7 2 8  
6 9 4 0 9 1 1 2 4  
3 2 7 3 8 6 9 0 5  
6 0 7 6 1 8 7 9 3  
9 8 5 9 3 3 0 7 4  
9 8 0 9 4 1 4 4 6  
0 4 5 6 1 0 0 1 7  
1 6 3 0 2 1 1 7 9

Training samples

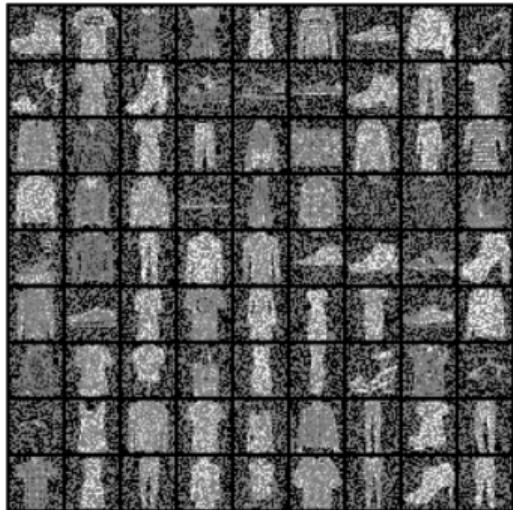
2 4 1 8 9 3 7 1 8  
7 5 8 3 0 9 0 2 4  
6 9 1 5 7 5 2 4 0  
8 3 7 4 8 0 1 2 8  
9 0 0 2 5 6 6 5 5  
8 8 3 8 9 0 9 3 9  
3 8 3 4 7 8 0 8 8  
2 0 8 4 2 8 9 8 3  
5 3 3 8 4 0 6 3 8

Generated samples

# Training from Incomplete and Noisy Data: Working Examples



Clean samples



Training samples

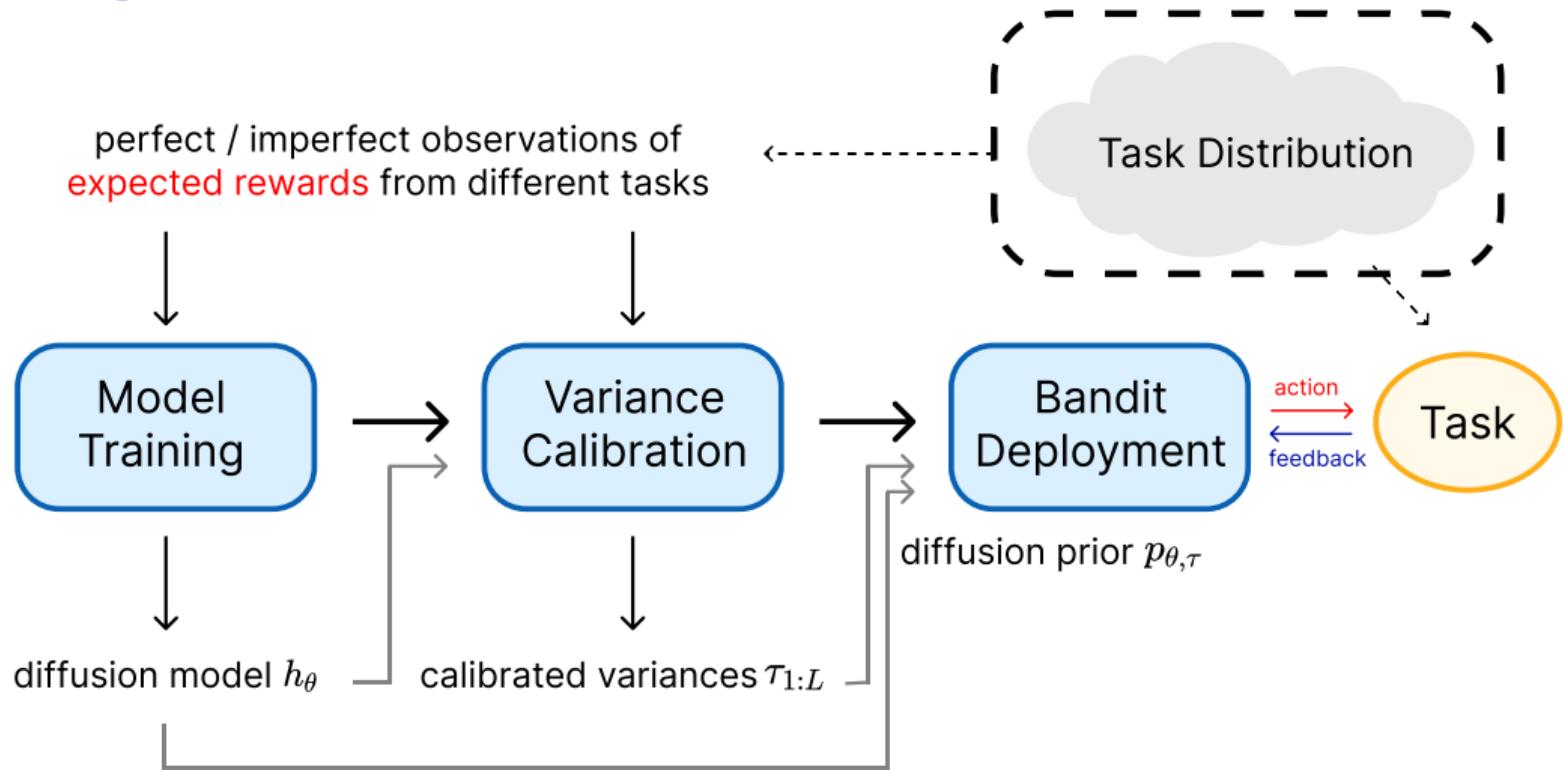


Generated samples

# Plan

- ① Multi-Armed Bandits and Meta-Learning
- ② Denoising Diffusion / Score-Based Models
- ③ Algorithms
- ④ Numerical Experiments
- ⑤ Conclusion and Perspectives

# Describing the Task Distribution



# Experimental Setup– Popular and Niche

Recommend items to customers

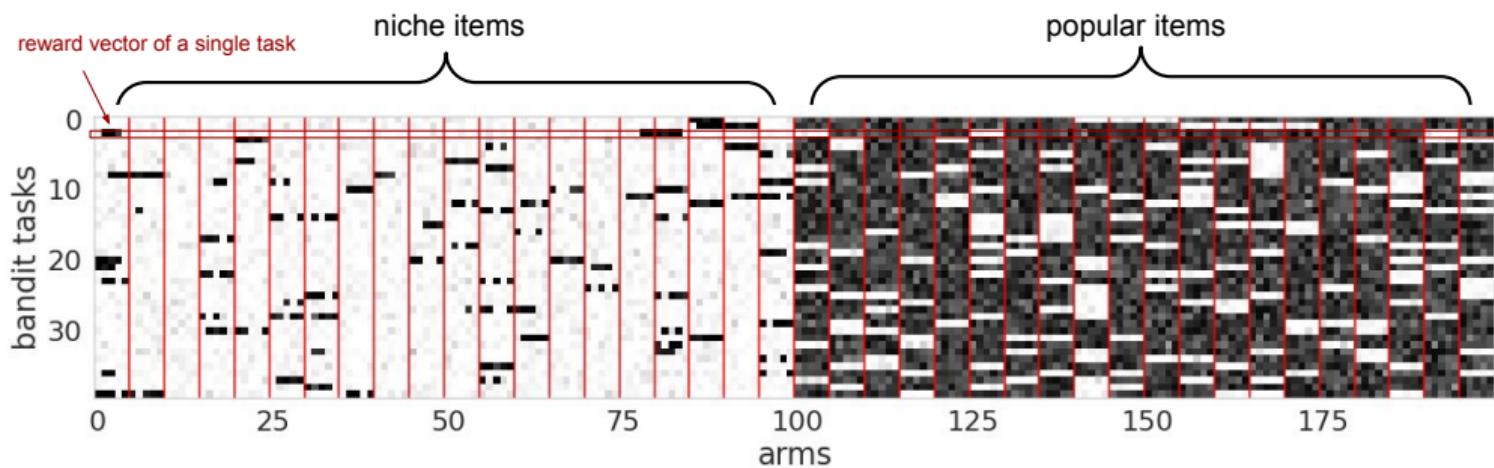
- Popular items: gift cards, electronics, clothing, ...
- Niche items: artworks, fan merch, ...

The screenshot shows the "Amazon Best Sellers" section on the Amazon homepage. It features three main sections: "Best Sellers in Toys & Games", "Best Sellers in Kitchen & Dining", and "Best Sellers in Clothing, Shoes & Jewelry". Each section displays a grid of products with images, titles, and price tags. The "Toys & Games" section includes items like a Rubik's cube, a train set, and a connect game. The "Kitchen & Dining" section includes items like a coffee maker, a slow cooker, and a rice cooker. The "Clothing, Shoes & Jewelry" section includes items like a hoodie, a beanie, and a dress.

The screenshot shows the "amazon merch on demand" website. It has a blue header with the site name and a navigation bar with links for "Resources" and "Sign in". Below the header is a large image of a workspace with a computer monitor displaying various designs, books, and decorative items. To the left of the image is a text block: "Become a Merch on Demand content creator. Share your designs with the world by creating graphic tees, accessories, and more, all printed on demand. We handle your printing and shipping, so you can design while we deliver." Below the image are five icons with text: "Upload your artwork!", "Set a list price", "We print what's sold", "Fast shipping with Prime", and "Gain monthly royalties". At the bottom, there are two sections: "How does it work?" with a yellow warning icon and "Want to join our community of content creators?", both with descriptive text and a "Sign up" button.

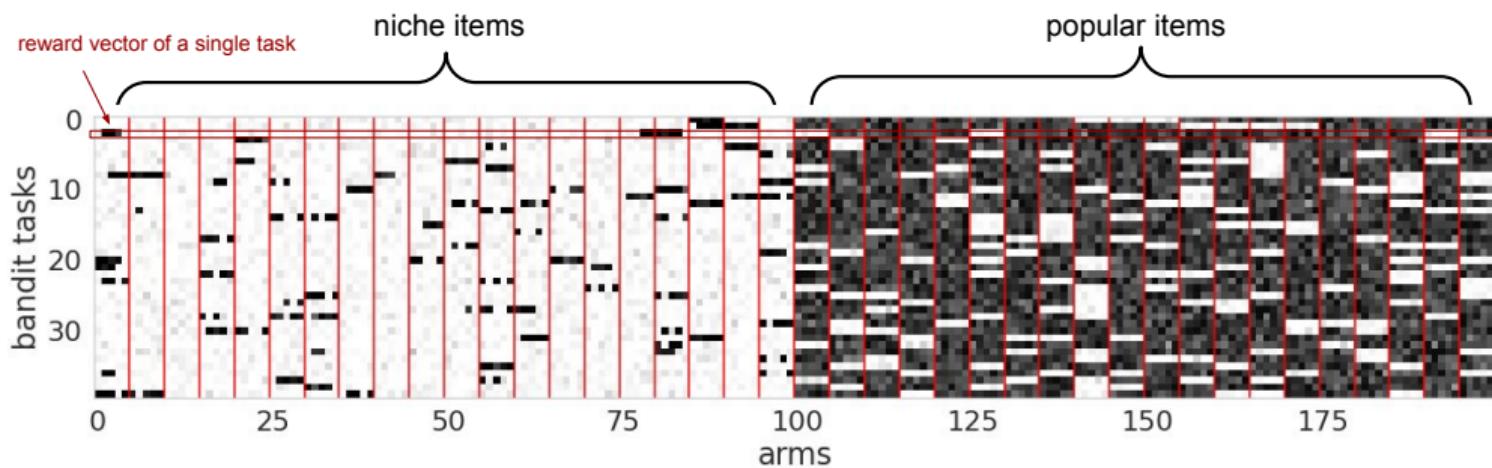
## Experimental Setup– Popular and Niche

- $K = 200$  arms (items)  $\mu \in [0, 1]^{200}$  are split into 40 groups with equal size



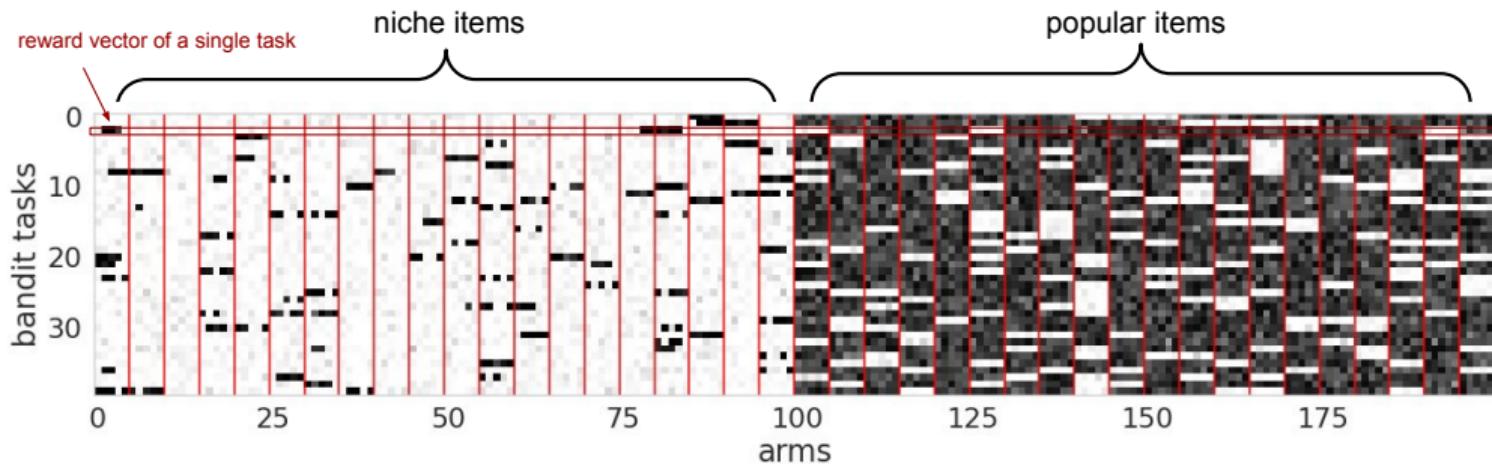
## Experimental Setup– Popular and Niche

- $K = 200$  arms (items)  $\mu \in [0, 1]^{200}$  are split into 40 groups with equal size
- 20 groups of arms represent the popular items that tend to have higher means



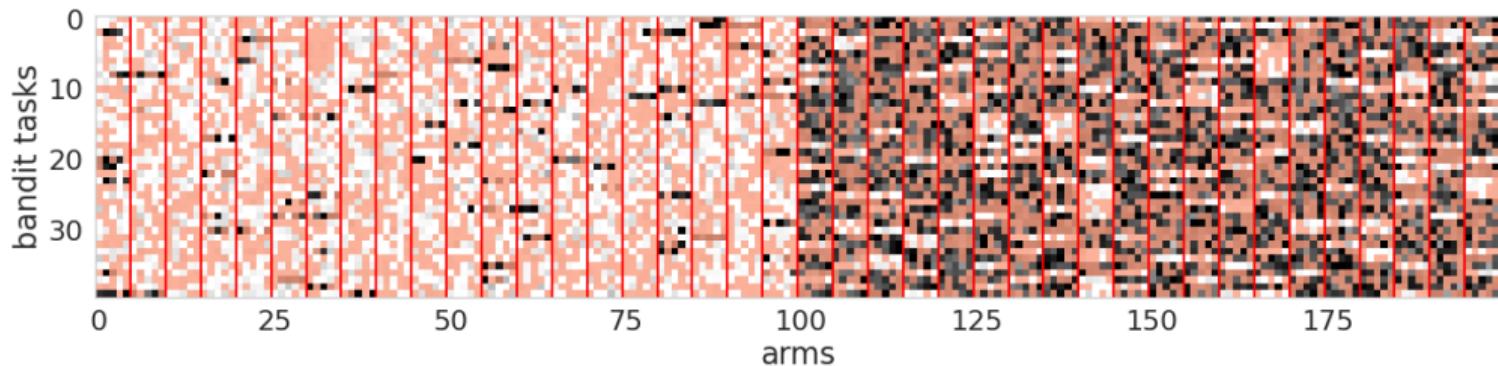
## Experimental Setup– Popular and Niche

- $K = 200$  arms (items)  $\mu \in [0, 1]^{200}$  are split into 40 groups with equal size
- 20 groups of arms represent the popular items that tend to have higher means
- 20 groups of arms represent the niche items that have lower means in general but some of these items get much higher means



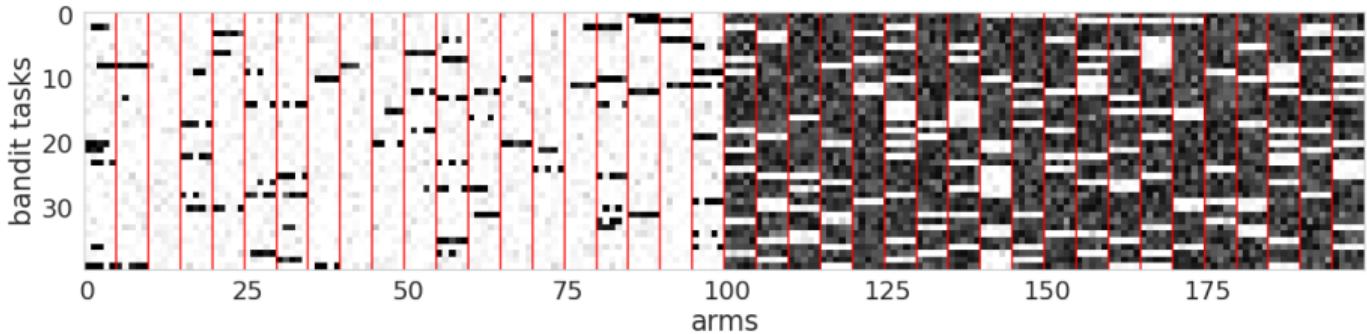
## Experimental Setup– Popular and Niche

- $K = 200$  arms (items)  $\mu \in [0, 1]^{200}$  are split into 40 groups with equal size
- 20 groups of arms represent the popular items that tend to have higher means
- 20 groups of arms represent the niche items that have lower means in general but some of these items get much higher means
- Imperfect data: noise with standard deviation 0.1 and missing rate 0.5

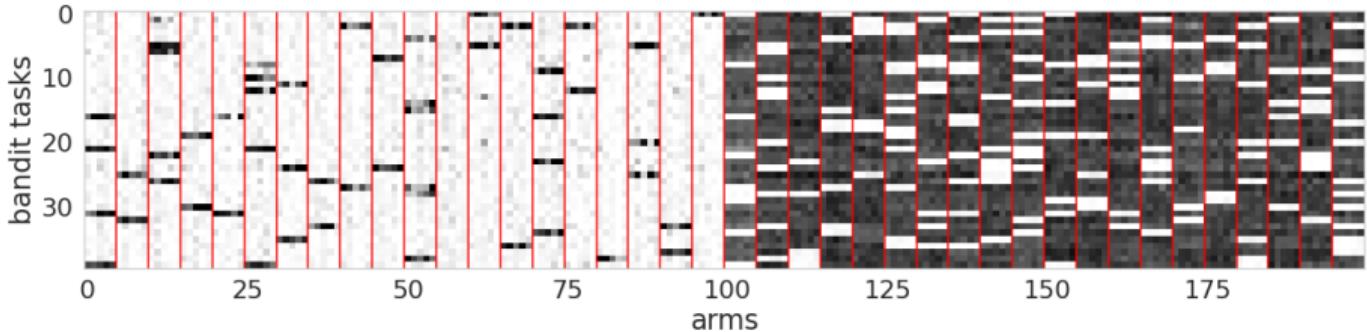


# Samples Generated by Learned Diffusion model

Perfect  
data

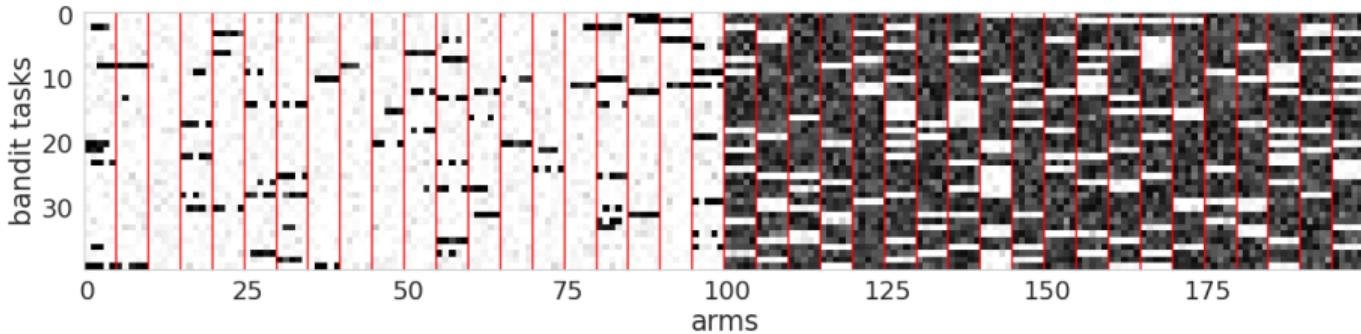


Generated  
Trained on  
clean data

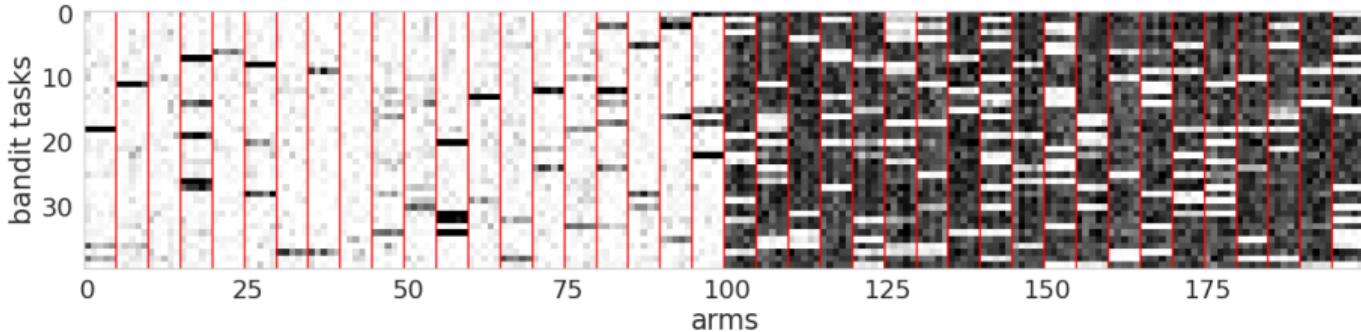


# Samples Generated by Learned Diffusion model

Perfect  
data



Generated  
Trained on  
incomplete  
noisy data



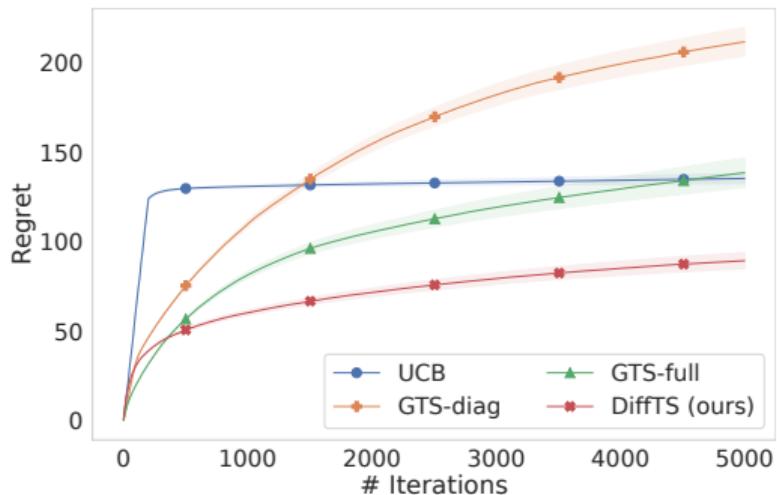
# Further Experimental Details

- Training set of size 5000; Calibration set of size 1000; Test on 100 tasks
- To generate reward add Gaussian noise with standard deviation 0.1
- Baselines: UCB, Thompson sampling with diagonal or full covariance Gaussian prior
- Gaussian mean and variance/covariance are fitted using the same perfect/corrupted training + calibration set
- Algorithms are run with groundtruth noise standard deviation 0.1

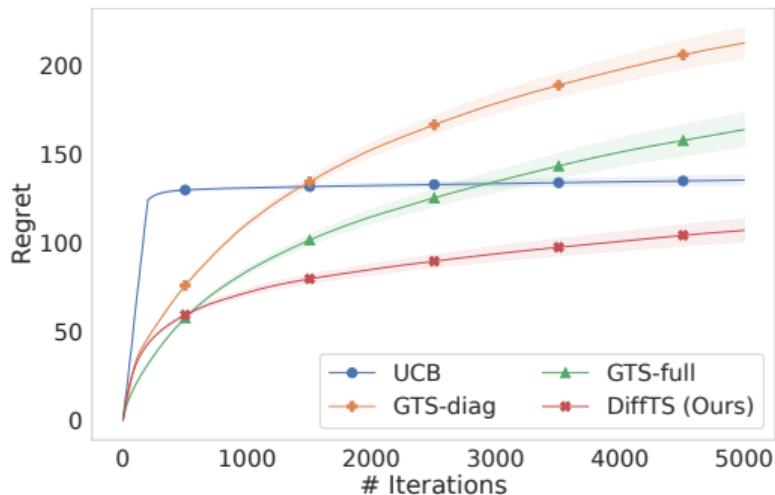
# Experimental Results

**Regret** is the cumulative difference between an algorithm and the one that consistently

pulls an optimal arm  $a^*$ :  $\text{Reg}_T = T\mu^{a^*} - \sum_{t=1}^T \mu^{a_t}$



Fit on perfect data



Fit on imperfect data

# Plan

- ① Multi-Armed Bandits and Meta-Learning
- ② Denoising Diffusion / Score-Based Models
- ③ Algorithms
- ④ Numerical Experiments
- ⑤ Conclusion and Perspectives

# Summary

- We propose to learn the prior of a **bandit** algorithm with **diffusion models** under the **meta-learning framework**
- We design a **Thompson sampling** algorithm to use the learned diffusion model that balances between prior and observations
- We design a **training** procedure to learn diffusion model from **incomplete** and **noisy** data
- We demonstrate the potential of our approach through several experiments

# Perspectives

- Contextual bandits → Distribution in function space
- Training with more complex missing mechanism (e.g., logged data) and general noise
- Theoretical justification of the benefit of the diffusion model
- Can we have a theoretically founded safeguard mechanism?

# Perspectives

- Contextual bandits → Distribution in function space
- Training with more complex missing mechanism (e.g., logged data) and general noise
- Theoretical justification of the benefit of the diffusion model
- Can we have a theoretically founded safeguard mechanism?

Thank you for your attention

---

## Algorithm Meta Learning for Bandits with Diffusion Models

---

- 1: Meta Training
  - 2: **Input:** Observations of expected rewards  $(\mu_B)_B$  from different tasks  $B \sim \mathcal{T}$
  - 3: Train a diffusion model  $h_\theta$  to model the distribution of the mean rewards
  - 4: Variance Calibration
  - 5: **Input:** Observations of expected rewards  $(\mu_B)_B$  from different tasks  $B \sim \mathcal{T}$
  - 6: Estimate the mean squared reconstruction error  $(\tau_\ell)_{\ell \in \{1, \dots, L\}}$  for the model  $h_\theta$  at different noise levels to calibrate the variance
  - 7: Meta Test / Deployment
  - 8: For any new task  $B$ , run Thompson sampling with the learned diffusion prior
-