Diffusion Prior for Online Decision Making
A Case Study of Thompson Sampling

Yu-Guan Hsieh

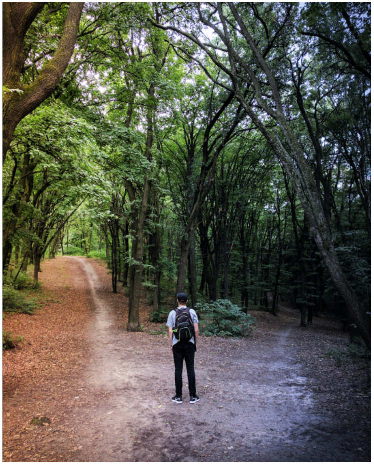
Supervisor: Shiva Kasiviswanathan

Mentor: Patrick Bloebaum

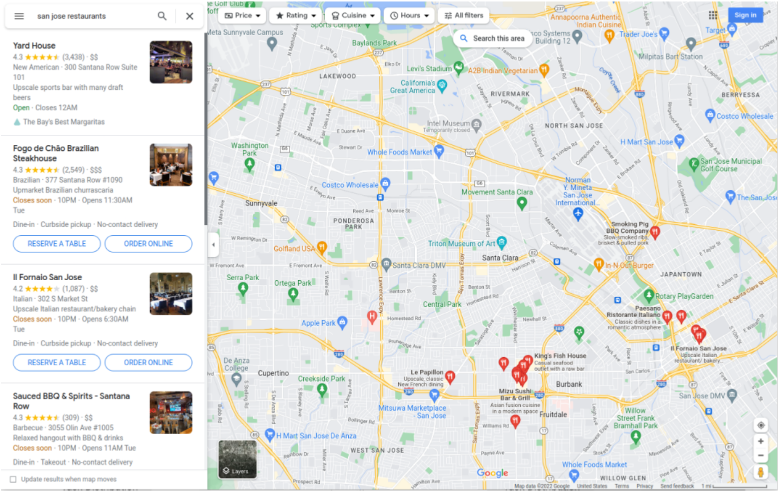
Also working with: Branislav Kveton

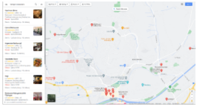
Internship from 08.01.2022 to 11.25.2022 in AWS causality team

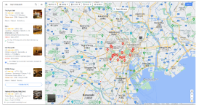
Uncertainty in Online Decision Making

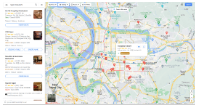


Prior Knowledge in Decision Making









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

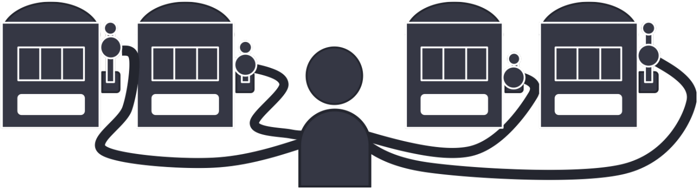
Explore online decision making with prior described by deep generative model

- 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

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

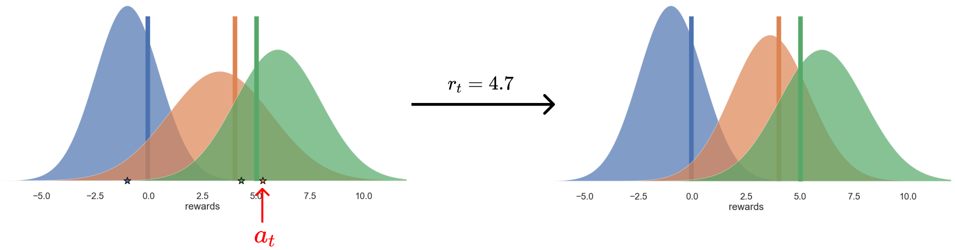
- A Bayesian approach to tackle multi-armed bandits
- The decision is random
- Has often better empirical performance than UCB (frequentist and deterministic)
- Precisely, for the parameter of interest w it maintains posterior distribution

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

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

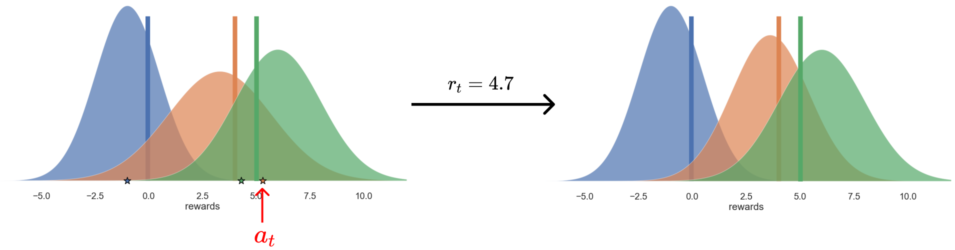
Thompson Sampling

- In vanilla MAB with 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

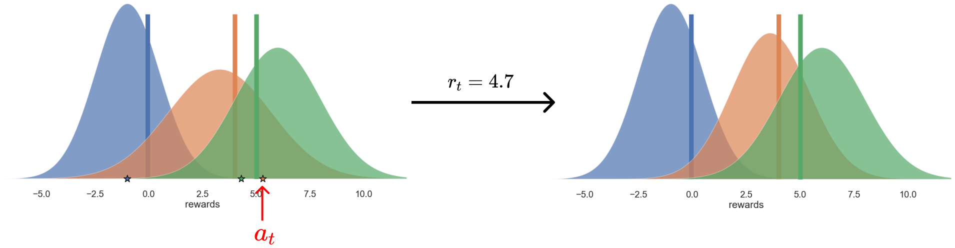
- The algorithm is sensitive to the choice of prior



Thompson Sampling

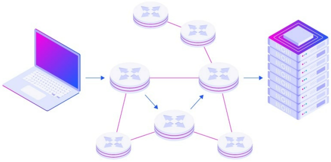
- The algorithm is sensitive to the choice of prior

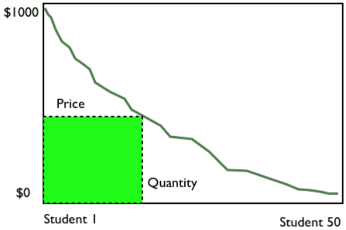
Can we learn the prior?



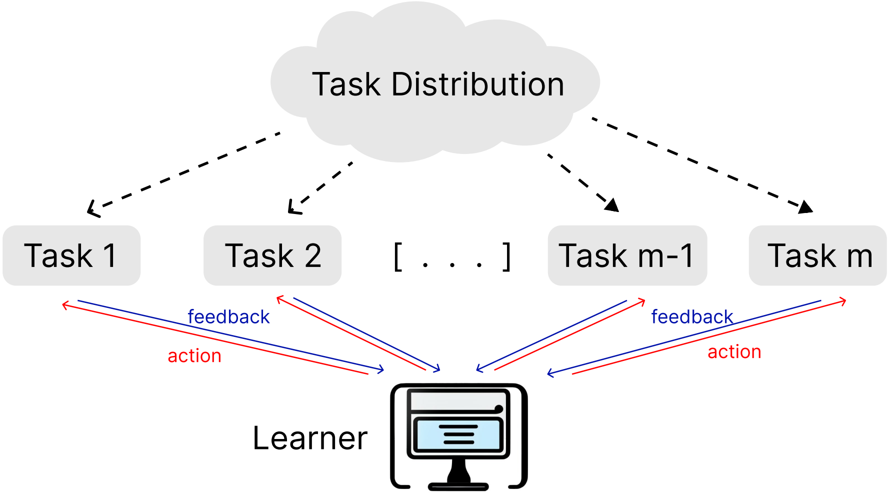
A Class of Bandit Tasks

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

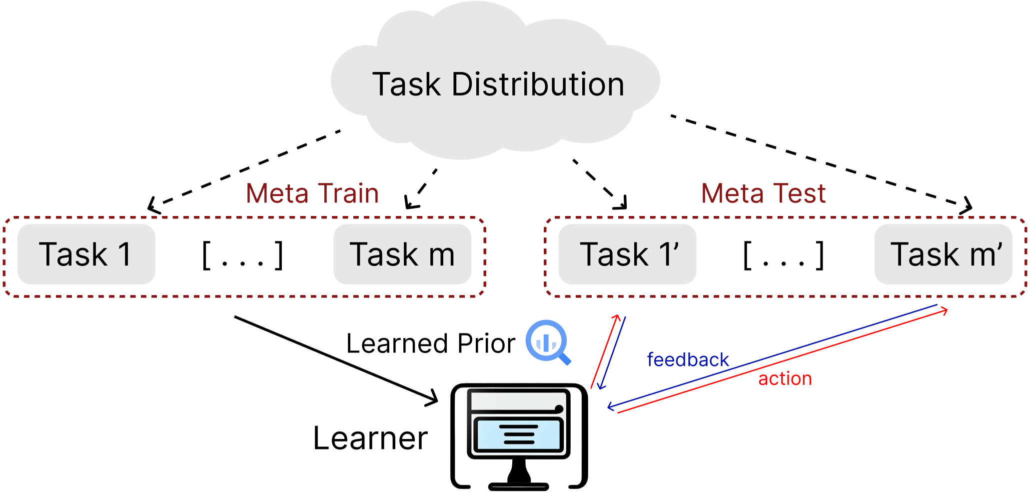




Meta Learning a Prior for Bandits



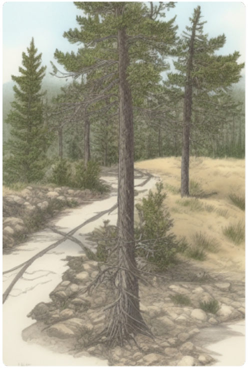
Meta Learning a Prior for Bandits

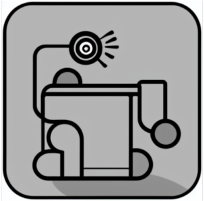


Plan

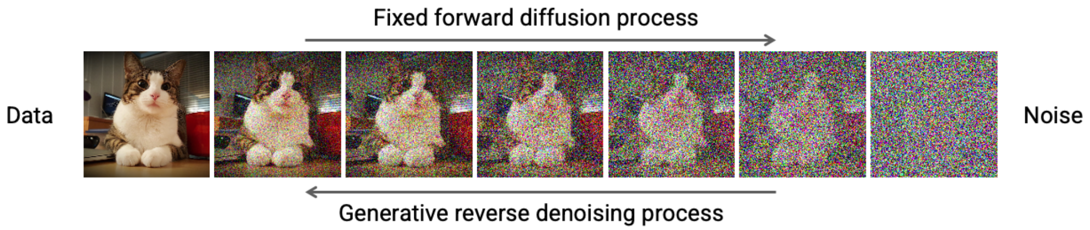
The Rise of Diffusion Models

- 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

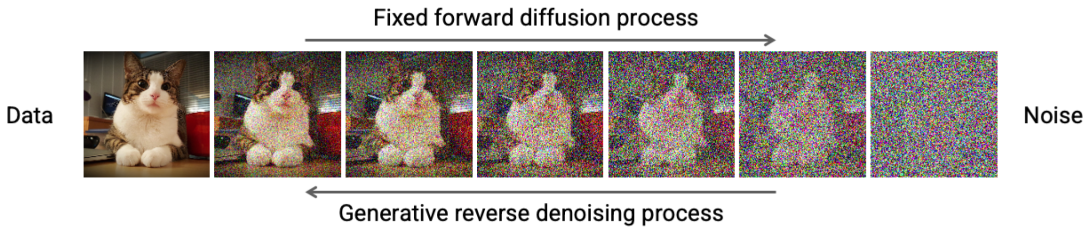


(Source: 2022 CVPR diffusion model tutorial)

- Add noise in the forward process: $q(X_{\ell+1} \mid x_\ell) = \mathcal{N}(X_{\ell+1}; \sqrt{\alpha_{\ell+1}} x_\ell, (1-\alpha_{\ell+1})I)$
- Parameterize the reverse process with a denoiser h_θ

$$p_\theta(X_\ell \mid x_{\ell+1}) = q(X_\ell \mid x_{\ell+1}, X_0 = h_\theta(x_{\ell+1}, \ell+1)) \propto \overbrace{q(x_{\ell+1} \mid X_\ell) q(X_\ell \mid X_0 = h_\theta(x_{\ell+1}, \ell+1))}^{\text{both are Gaussian by construction}}$$

Diffusion Models in a Nutshell



(Source: 2022 CVPR diffusion model tutorial)

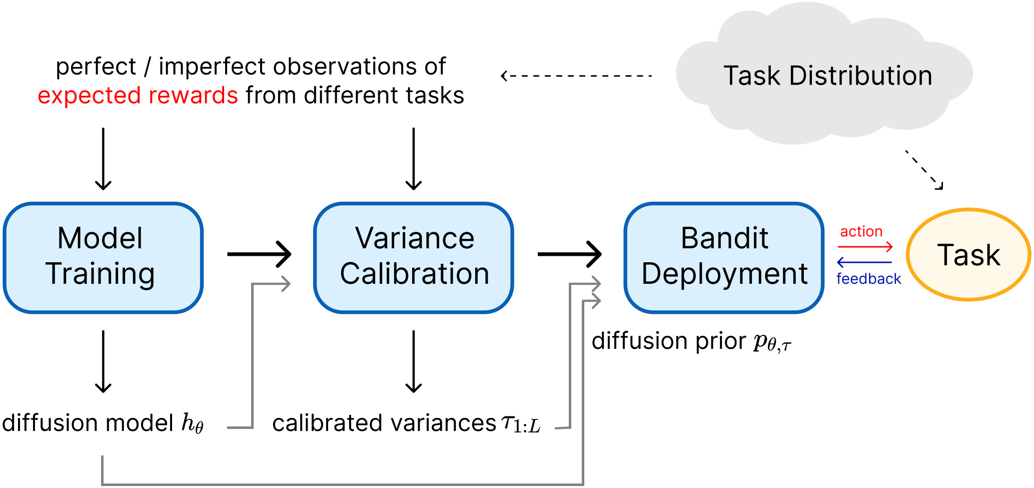
- The denoiser is trained to 'denoise'
- Diffusion model as maximum likelihood estimation / reverse-time SDE
- **The iterative sampling process allows for better posterior sampling**

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?

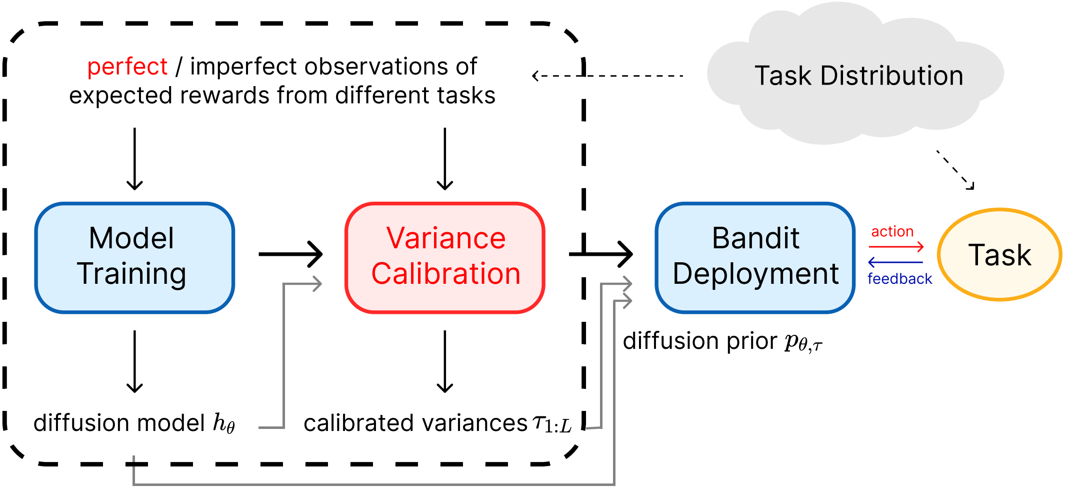
Plan

Overview


perfect / imperfect observations of
expected rewards from different tasks
Task Distribution
Model Training
Variance Calibration
Bandit Deployment
action
feedback
Task
diffusion prior $p_{\theta,\tau}$
diffusion model h_θ
calibrated variances $\tau_{1:L}$

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 $\mathrm{diag}(\tau^2_{\ell+1})$
- τ^2 is the mean squared reconstruction error $\tau^a_\ell = \sqrt{\mathbb{E}_{X_0, X_\ell}[\|X^a_0 - h^a_\theta(X_\ell, \ell)\|^2]}$

Variance Calibration

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

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

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

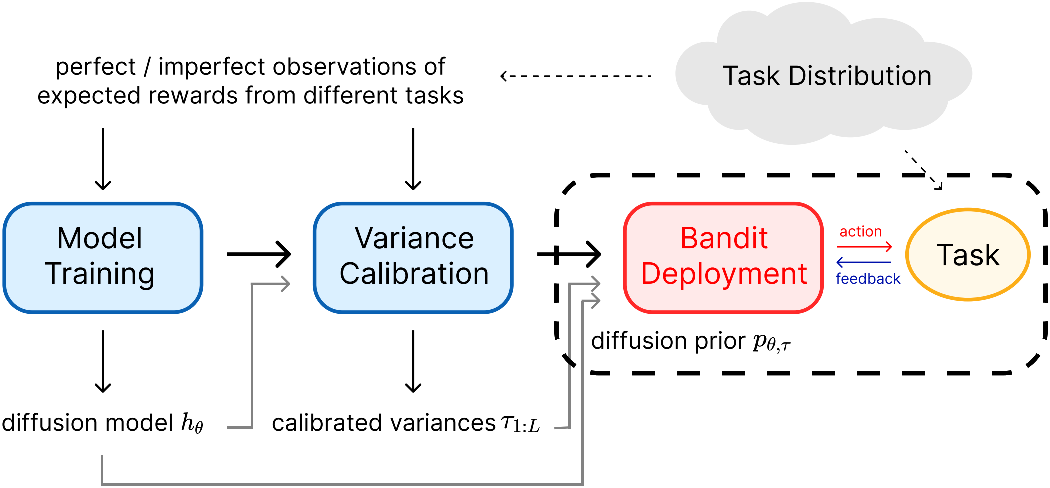
Variance Calibration

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

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

- τ^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 τ^2 from incomplete and noisy data

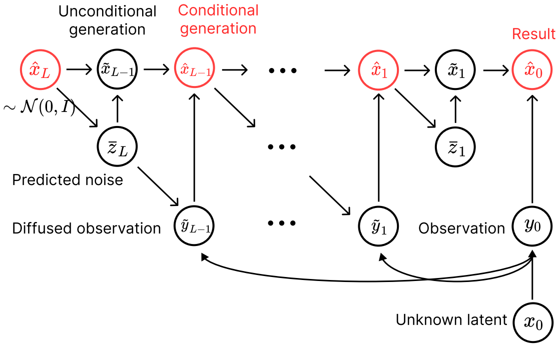
Thompson Sampling with Diffusion Prior



Thompson Sampling with Diffusion Prior

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

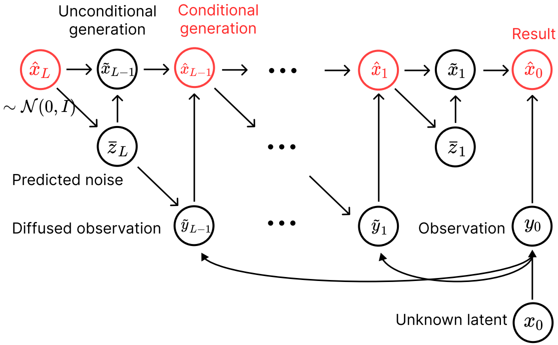
- In MAB, $y_0 = \mathcal{H}$ is the history



Thompson Sampling with Diffusion Prior

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

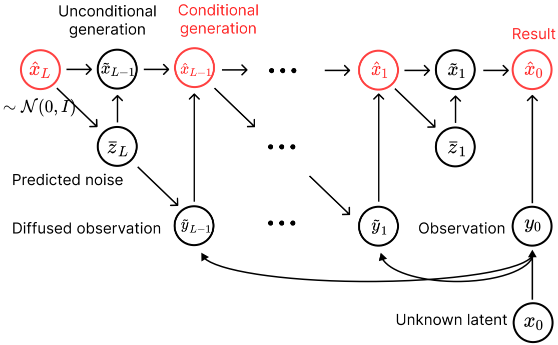
- In MAB, $y_0 = \mathcal{H}$ is the history
- Condition the reverse process on y_0
  - Sample x_L from $X_L \,|\, y_0$
  - Sample x_ℓ from $X_\ell \,|\, x_{\ell+1}, y_0$



Thompson Sampling with Diffusion Prior

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

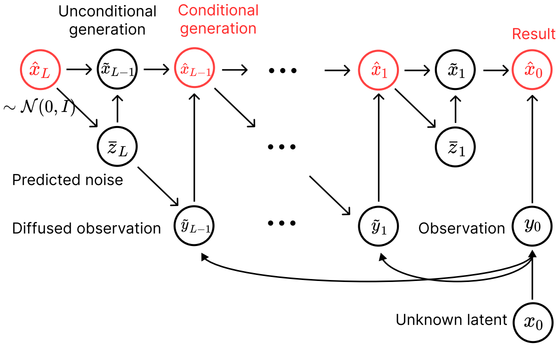
- In MAB, $y_0 = \mathcal{H}$ is the history
- Condition the reverse process on y_0
  - Sample x_L from $X_L \mid y_0$
  - Sample x_ℓ from $X_\ell \mid x_{\ell+1}, y_0$
- Initialization: Sampled from $\mathcal{N}(0, I)$



Thompson Sampling with Diffusion Prior

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

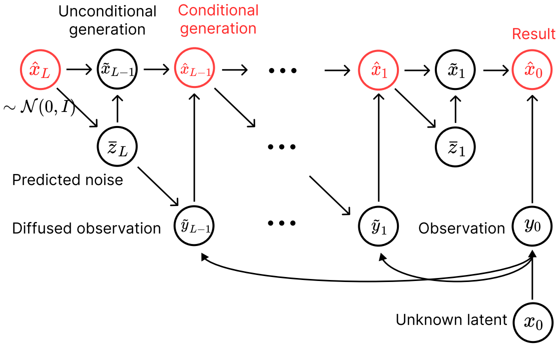
- In MAB, $y_0 = \mathcal{H}$ is the history
- Condition the reverse process on y_0
  - Sample x_L from $X_L \,|\, y_0$
  - Sample x_ℓ from $X_\ell \,|\, x_{\ell+1}, y_0$
- Initialization: Sampled from $\mathcal{N}(0, I)$
- Recursion: Mix an unconditional sampled $\tilde{x}_\ell$ with a *diffused* $\tilde{y}_\ell$



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_ℓ 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* $\tilde{y}_\ell$ → 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 $p_{\theta,\tau}(X_0 \mid y_0)$ provided imperfect observation y_0

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

Set $\tilde{q}(x^a_\ell \mid x_{\ell+1}, y_0) \propto \underbrace{p_{\theta,\tau}(x^a_\ell \mid x_{\ell+1})}_{\text{prior}} \mathcal{N}\Big(x^a_\ell ; \underbrace{\tilde{y}^a_\ell}_{\text{observation}}, \bar{\alpha}_\ell\big((\sigma^a_t)^2 + \underbrace{\rho_\ell(\tau^a_{\ell+1})^2}_{\text{denoising variance}}\big)\Big)$

Algorithm Thompson Sampling with Diffusion Prior (DiffTS)

Input: Trained denoiser h_θ, denoising variance $(\tau_\ell^2)_{\ell \in \{1,\ldots,L\}}$, presumed noise std σ'
for $t = 1, \ldots$ **do**

Posterior Sampling

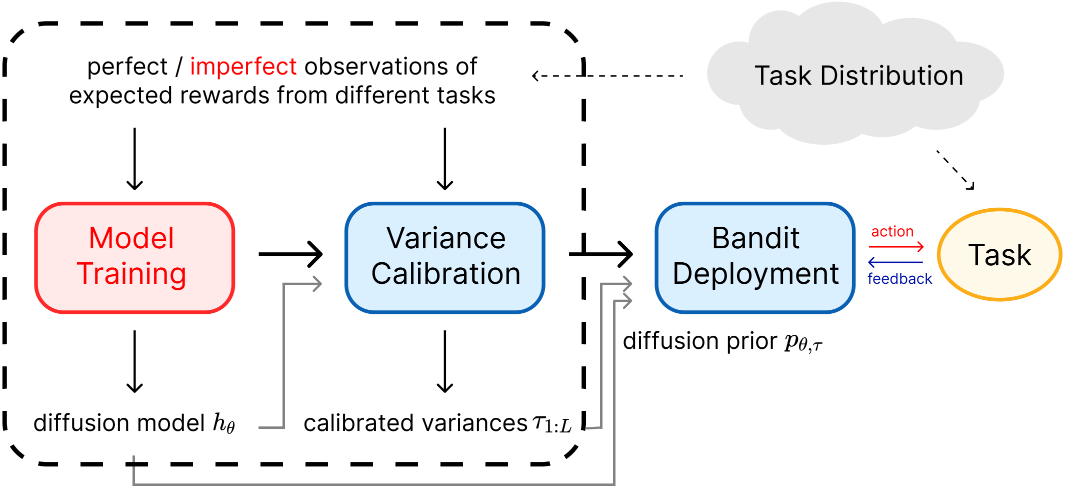
  Sample $x_L \sim \mathcal{N}(0, I)$
  for $\ell \in L-1, \ldots, 0$ **do**
    Predict clean sample $\hat{x}_0 = h_\theta(x_{\ell+1}, \ell+1)$ and associated noise $\bar{z}_{\ell+1}$
    Compute diffused observation $\tilde{y}_\ell^a = \sqrt{\bar{\alpha}_\ell}\hat{\mu}_{t-1}^a + \sqrt{1-\bar{\alpha}_\ell}\bar{z}_{\ell+1}$
    for $a \in \mathcal{A}$ **do**
      If $N_{t-1}^a = 0$, sample $x_\ell^a \sim p_{\theta,\tau}(X_\ell^a \,|\, x_{\ell+1})$
      If $N_{t-1}^a > 0$, sample

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

  Pull arm $a_t \in \arg\max_{a \in \mathcal{A}} x_0^a$
  Update number of pulls N_t^a, scaled std σ_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 $\mathbb{E}_{\ell,X_0,X_\ell}[\|X_0 - h_\theta(X_\ell,\ell)\|^2]$

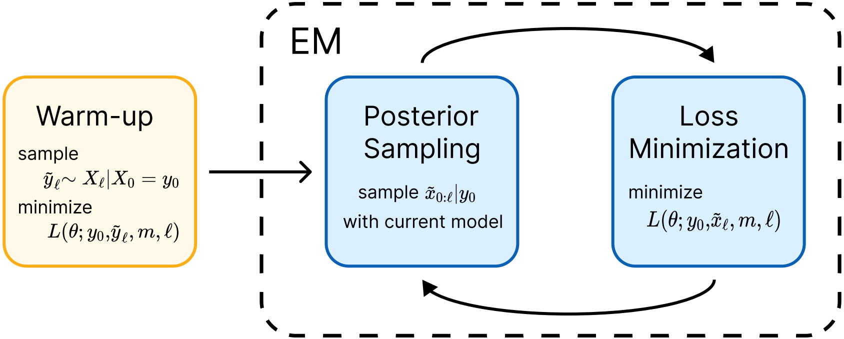
- Training from perfect data x_0: minimize standard diffusion loss

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

- 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_ℓ are not available

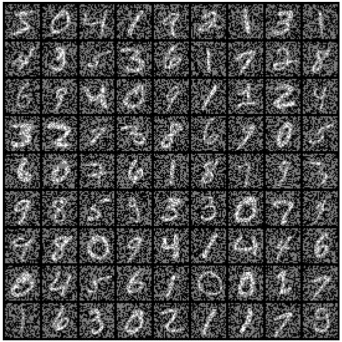
Training from Incomplete and Noisy Data



$$L(\theta; y_0, \tilde{x}_\ell, m, \ell) = \underbrace{\| m \odot y_0 - m \odot h_\theta(\tilde{x}_\ell, \ell) \|^2}_{\text{ignored masked value}} + \underbrace{2\lambda \sqrt{\bar{\alpha}_\ell}\sigma^2 \, \mathbb{E}_{b \sim \mathcal{N}(0,I)} \, b^\top \left(\frac{h_\theta(\tilde{x}_\ell + \varepsilon b, \ell) - h_\theta(\tilde{x}_\ell, \ell)}{\varepsilon} \right)}_{\text{MC-SURE regularization to counter noise}}$$

Training from Incomplete and Noisy Data: Working Examples

Clean samples

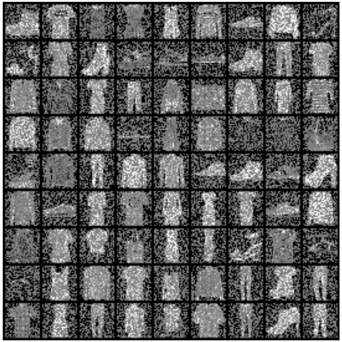


Training samples

Generated samples

Training from Incomplete and Noisy Data: Working Examples

Clean samples

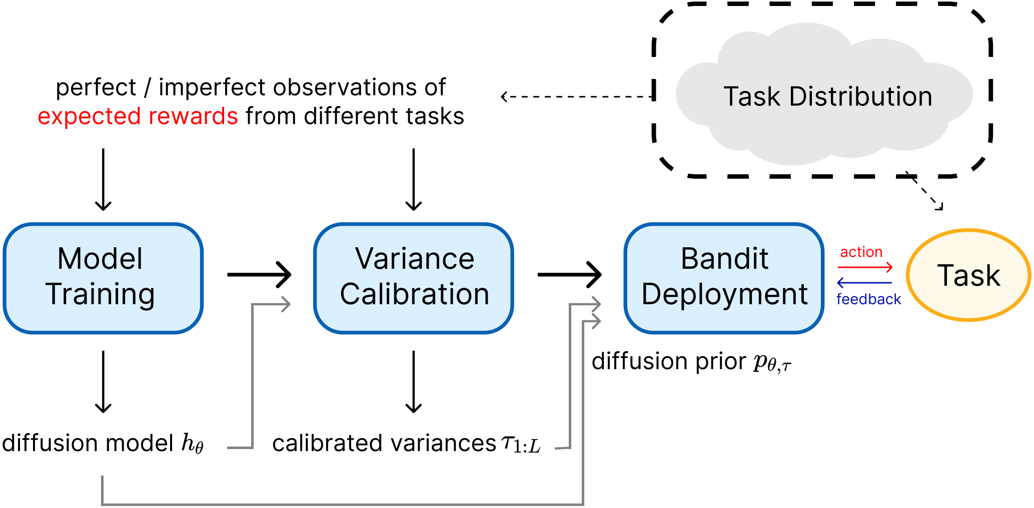


Training samples

Generated samples

Plan

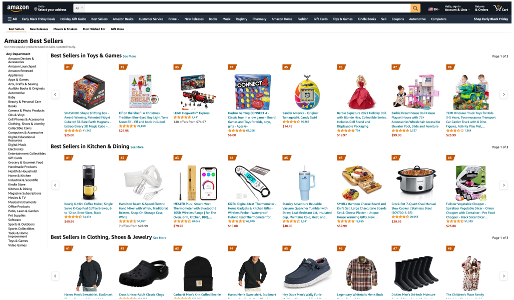
Describing the Task Distribution

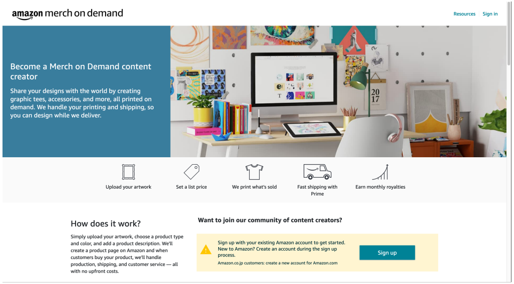


Experimental Setup– Popular and Niche

Recommend items to customers

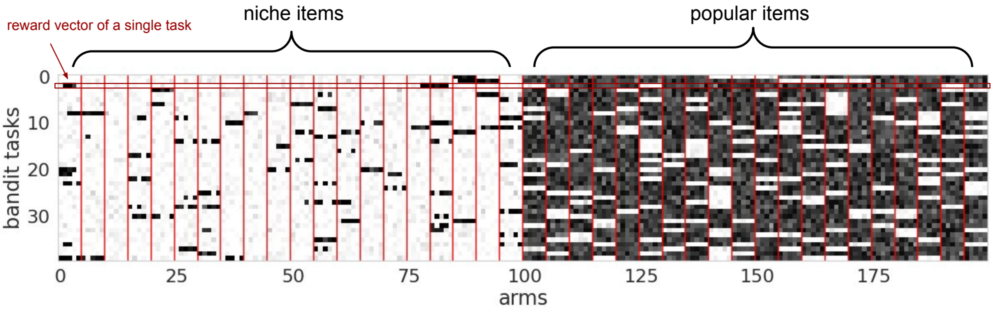
- Popular items: gift cards, electronics, clothing, . . .
- Niche items: artworks, fan merch, . . .





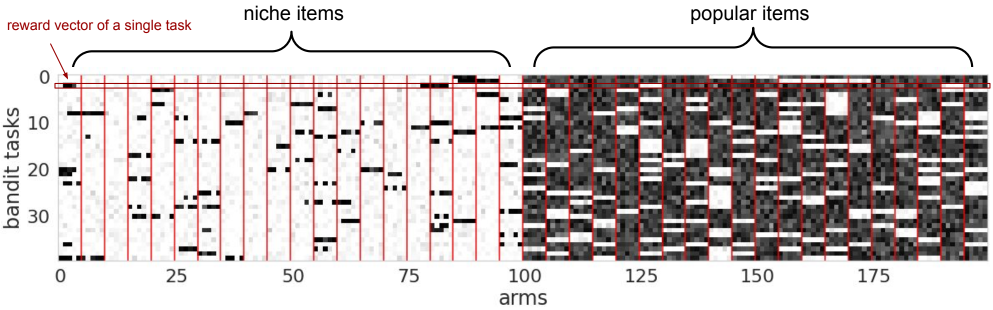
Experimental Setup– Popular and Niche

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



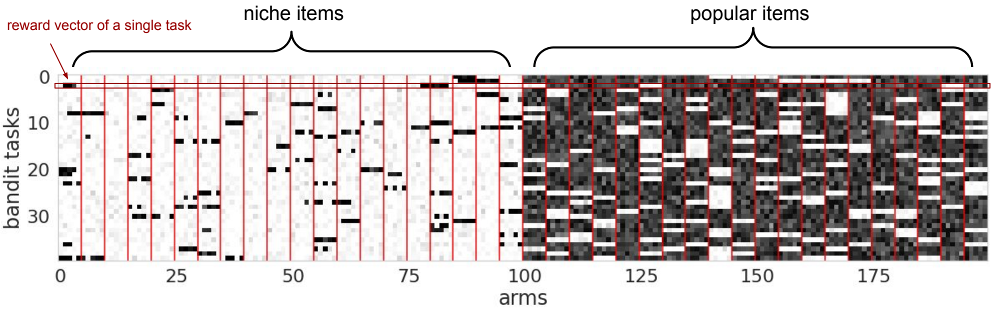
Experimental Setup– Popular and Niche

- $K = 200$ 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

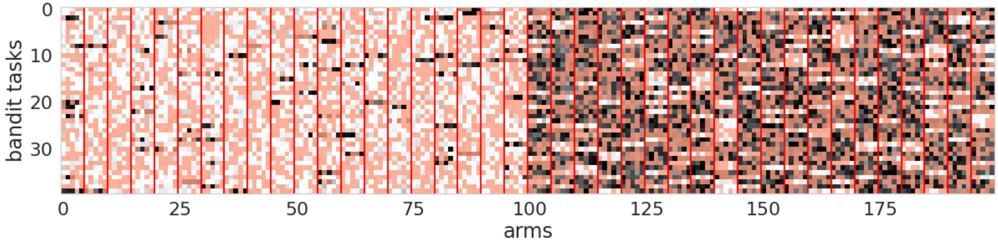
- $K = 200$ arms (items) $\mu \in [0,1]^{200}$ are split into 40 groups with equal size
- 20 groups of arms represent the popular items that tend to have higher means
- 20 groups of arms represent the niche items that have lower means in general but some of these items get much higher means



Experimental Setup– Popular and Niche

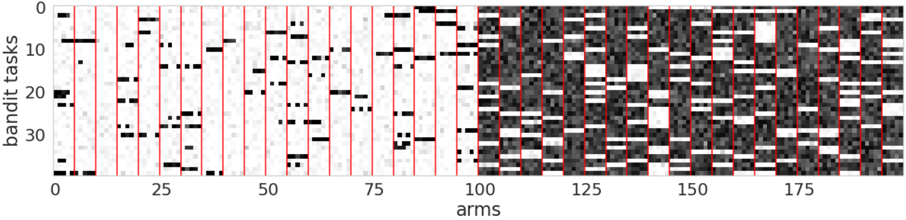
- $K = 200$ arms (items) $\mu \in [0,1]^{200}$ are split into 40 groups with equal size
- 20 groups of arms represent the popular items that tend to have higher means
- 20 groups of arms represent the niche items that have lower means in general but some of these items get much higher means

- Imperfect data: noise with standard deviation 0.1 and missing rate 0.5



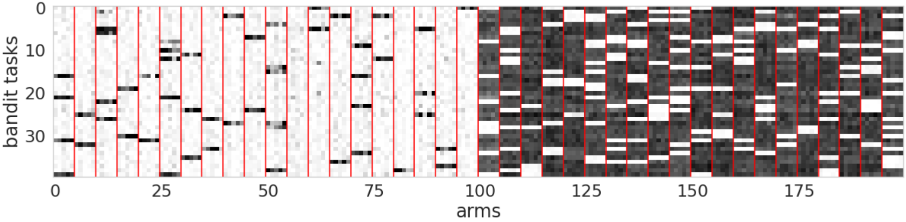
Samples Generated by Learned Diffusion model

Perfect data



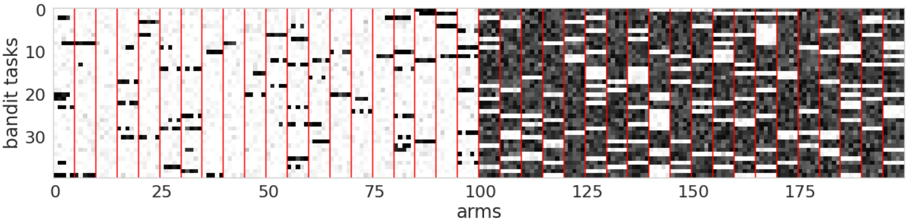
Generated

Trained on clean data



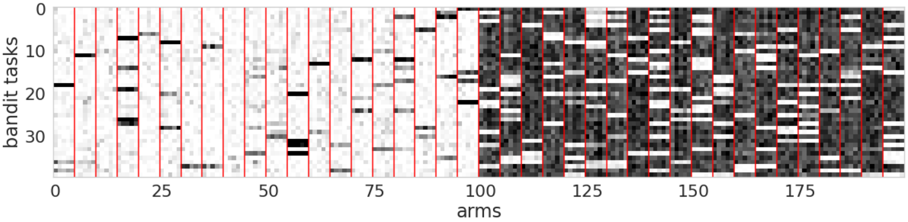
Samples Generated by Learned Diffusion model

Perfect data



Generated

Trained on incomplete noisy data

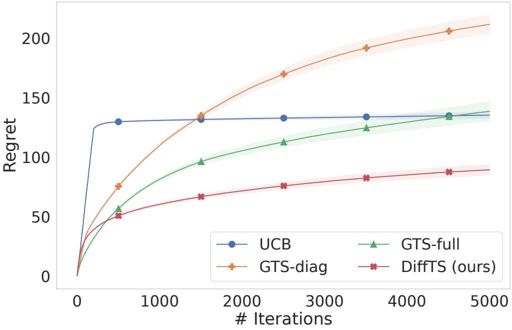


Further Experimental Details

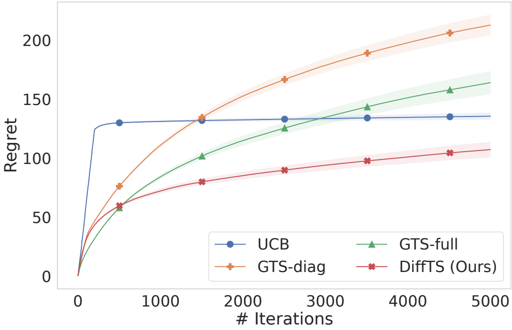
- Training set of size 5000; Calibration set of size 1000; Test on 100 tasks
- To generate reward add Gaussian noise with standard deviation 0.1
- Baselines: UCB, Thompson sampling with diagonal or full covaraince Gaussian prior
- Gaussian mean and variance/covariance are fitted using the same perfect/corrupted 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 $a^\star$: $\mathrm{Reg}_T = T\mu^{a^\star} - \sum_{t=1}^{T} \mu^{a_t}$



Fit on perfect data



Fit on imperfect data

Plan

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

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