Quantum Generative Modelling

Rajit Rajpal, Uzay Karadag, James Zoryk
Supervisor: Prof. Benedict Leimkuhler

Generative Modelling

Background

- **Have**: One collection of samples X from unknown distribution P
- **Goal**: Generate samples Y that look like P
- **Why?**: The explosion in applications of generative AI today e.g. LLMs like ChatGPT, Image generation using Stable Diffusion, Audio generation models like WaveNet.
- **How?** Drive samples Y from a tuneable distribution Q to look like X by minimizing some cost function.

The Cost Function: Maximum Mean Discrepancy

- The Maximum Mean Discrepancy (MMD) is used as a cost function by computing the difference in means of the distributions lifted to feature space H_k.

$$\begin{aligned} MMD^2(P, Q) &= \|\mu_P - \mu_Q\|^2_{H_k} \\ &= \langle \mu_P, \mu_P \rangle_{H_k} + \langle \mu_Q, \mu_Q \rangle_{H_k} - 2\langle \mu_P, \mu_Q \rangle_{H_k} \\ &= \boldsymbol{E}_P[k(X, X')] + \boldsymbol{E}_Q[k(Y, Y')] + \boldsymbol{E}_{P,Q}[k(X, Y)] \end{aligned}$$

- $k(z, z') = e^{-\frac{(z-z')^2}{2\sigma^2}}$ is Gaussian kernel with bandwidth σ.
- $MMD^2(P, Q) = 0$ when $P = Q$

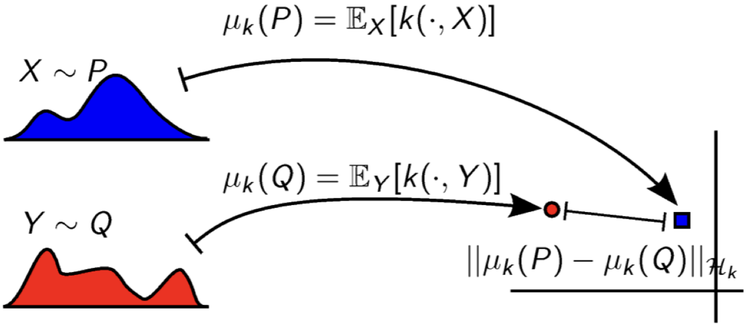


Figure 1: Maximum Mean Discrepancy

Introduction to Quantum Bayesian Learning

- Consider parameters $\boldsymbol{\theta}$ as random variables with a probability distribution $\pi(\boldsymbol{\theta}|\boldsymbol{D})$ and data $\boldsymbol{D}$:

$$\pi(\boldsymbol{\theta}|\boldsymbol{D}) \propto p(\boldsymbol{\theta})[Prior] f(\boldsymbol{D}|\boldsymbol{\theta})[Likelihood] = p(\boldsymbol{\theta}) \exp(-\beta C(\boldsymbol{\theta}))$$

- The goal becomes sampling $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N \sim \pi(\boldsymbol{\theta}|\boldsymbol{D})$ to approximately compute predictive distribution $p(y|\boldsymbol{D}) = \int_{\boldsymbol{\theta} \in \mathbb{R}^n} p(y|\boldsymbol{D}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{D})\, d\boldsymbol{\theta} \approx \frac{1}{N} \Sigma_{i=1}^{N} p(y|\boldsymbol{\theta}_i)$

- Working in log-space, we use gradient based stepping methods (e.g. Langevin Monte Carlo):

$$\nabla_{\boldsymbol{\theta}} \log \pi(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) - \beta \nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}).$$

Introduction to Parameterised Quantum Circuits

- Parameterised Quantum Circuits (PQCs) serve as the backbone of quantum algorithms.
- A typical PQC takes the form:

$$U(\boldsymbol{\theta}) = \prod_{k=1}^{K} W_k U_k([\boldsymbol{\theta}]_k),$$

  where $\{W_k\}$ are fixed quantum gates and $\{U_k([\boldsymbol{\theta}]_k)\}$ are parameterised gates.

We can map the problem to a cost function to minimize (or sample) using a parameterised quantum circuit (Ansatz).

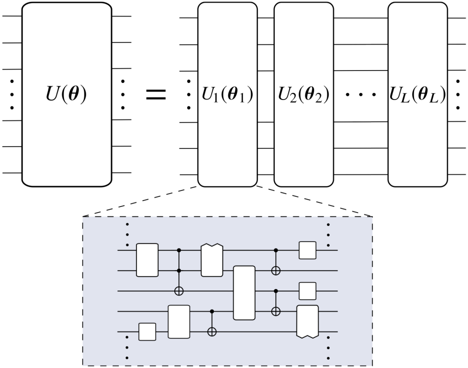


Figure 2: Schematic diagram of an ansatz *[Duffield et al. 2022]*

Introduction to Parameterised Quantum Circuits

Shift to Notebook

Born Machine

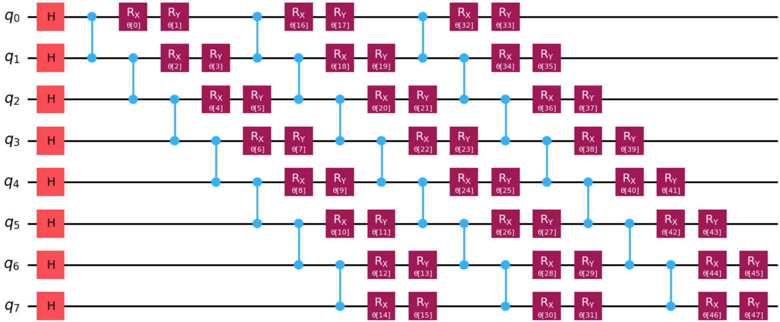


Figure 3: Born Machine with 8 qubits and a circuit depth of 3 thus giving us an Ansatz of 64 parameters.

Overview of Bayesian Learning using a PQC

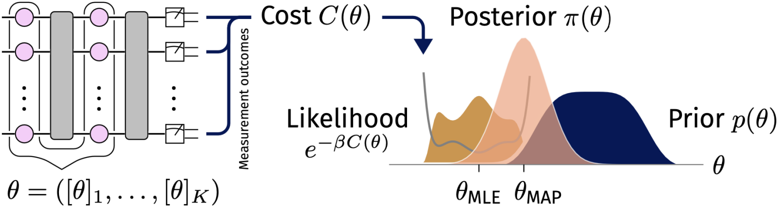


Figure 4: Overview of Quantum BL *[Duffield et al. 2022]*

Exploring the Posterior

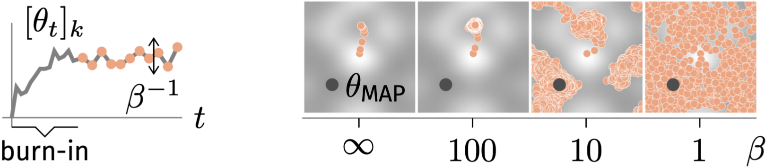


Figure 5: How to Explore the Posterior using Langevin Dynamics
[Duffield et al. 2022]

- Goal: Improving the representation power and sampling of complex distributions *[Biamonte et al., 2018]*.
- How?: Using a standard PQC (Born Machine) as an Ansatz [1] for the Posterior and driving the samples from the Born machine to be as close as possible to the empirical data distribution i.e., $\pi(\boldsymbol{\theta}|D) \propto f(\boldsymbol{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = e^{-\beta C(\boldsymbol{\theta})}.\mathbf{1}$, where $C(\boldsymbol{\theta})$ is the Maximum Mean Discrepancy.

[1] Ansatz: Assuming a particular structure for the solution and optimizing based on that. E.g Assume a polynomial ansatz $y(x) = ax^2 + bx + c$ for the solution of the ODE $\frac{d^2y}{dx^2} - 4y = 0$ and optimize w.r.t a, b, c.

Stochastic Gradient Langevin Dynamics (SGLD)

SGLD *[Welling and Teh, 2011]* is based on the following stochastic differential equation:

$$
\begin{aligned}
d\boldsymbol{\theta} &= -\frac{1}{\gamma}\nabla\tilde{C}(\boldsymbol{\theta})\,dt + \sqrt{\frac{2}{\gamma\beta}}\,d\boldsymbol{W} \\
&= -\frac{1}{\gamma}\left(\frac{1}{|S|}\sum_{i\in S}\nabla C_i(\boldsymbol{\theta})\right)dt + \sqrt{\frac{2}{\gamma\beta}}\,d\boldsymbol{W}
\end{aligned}
$$

where $S \subseteq 1, \ldots, n$ is a random subset of indices.

- $\boldsymbol{\theta}$: circuit parameters
- γ: friction coeffiicent
- $\tilde{U}(\boldsymbol{\theta})$: approximated potential
- β: reciprocal temperature
- $d\boldsymbol{W}$: standard Wiener process

Gradient Noise Model for GMM

Full gradients are costly to compute so we utilize stochastic gradients (Batch Size $|S_j|$= 200) which leads to bias:

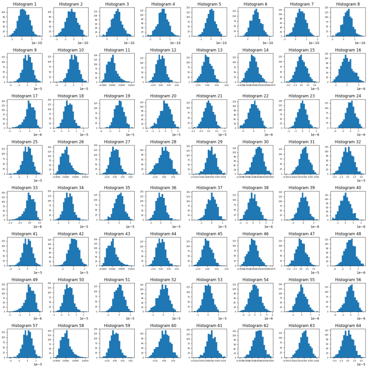


Figure 6: Component-Wise (Marginal) Distributions of bias vector $h_i = \{(\nabla \tilde{C}_{|S_j|}(\boldsymbol{\theta}) - \nabla C(\boldsymbol{\theta}))_i\}_{j=1}^{2000}$, $i = 1, ..., 64$

Enhanced Image of GNM

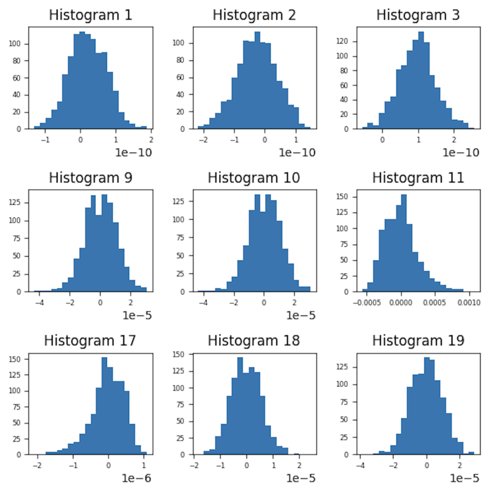


Figure 7: Enhanced Subset of Marginal Distributions

Methods to unbias

- Stepsize Adaptation *[Ryckaert and Ciccoti, 1977]*
- Multilevel Monte Carlo *[Chada et al, 2023]*
- Adaptive Langevin Thermostat *[Leimkuhler and Jones, 2011]*.

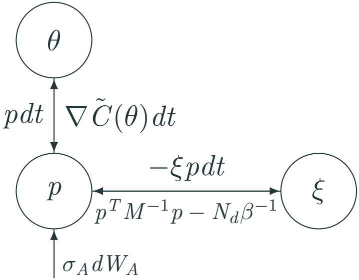
Adaptive Langevin (Ad-Langevin) Thermostat

The Ad-Langevin Thermostat *[Leimkuhler and Shang, 2016]* debiases under the assumption of Gaussian batch noise with constant covariance. It is based on the following coupled system of SDEs:

$$d\boldsymbol{\theta} = \boldsymbol{M}^{-1}\boldsymbol{p}\,dt$$
$$d\boldsymbol{p} = \nabla\tilde{C}(\boldsymbol{\theta})\,dt - \xi\boldsymbol{p}\,dt + \sigma_A\sqrt{\boldsymbol{M}}\,d\boldsymbol{W_A}$$
$$d\xi = \mu^{-1}\left[\boldsymbol{p}^T\boldsymbol{M}^{-1}\boldsymbol{p} - N_d\beta^{-1}\right]dt$$

- $\boldsymbol{\theta}$: circuit parameters
- $\boldsymbol{p}$: momentum variable
- $\boldsymbol{M}$: mass matrix
- $\nabla\tilde{C}(\boldsymbol{\theta})$: stochastic gradient of the cost function
- ξ: auxillary variable
- σ_A: additive noise
- $d\boldsymbol{W_A}$: standard Wiener process
- μ: thermal mass
- N_d: number of degrees of freedom
- $\beta^{-1} = k_B T$: reciprocal temperature

The Ad-Langevin Thermostat Illustrated



Key interactions:

- ξ: friction on $\boldsymbol{p}$,
- Additive noise, σ_A, injects heat,
- ξ dynamics driven by kinetic energy difference $(\boldsymbol{p}^T \boldsymbol{M}^{-1} \boldsymbol{p} - N_d \beta^{-1})$.

Negative feedback loop enables adaptive noise dissipation.

Ad-Langevin Thermostat, Splitting Method

- A novel approach utilising Ad-Langevin Thermostat to remove bias from noisy gradients. *[Leimkuhler and Shang, 2016]*:

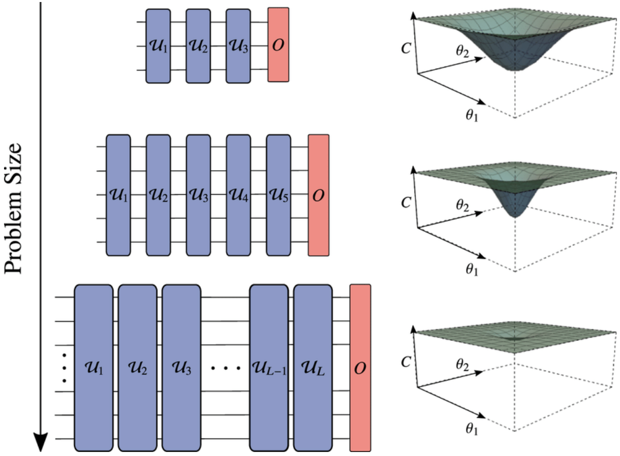
$$\mathrm{d}\begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{p} \\ \xi \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{M}^{-1}\boldsymbol{p} \\ \mathbf{0} \\ 0 \end{bmatrix} \mathrm{d}t}_{\mathrm{A}} + \underbrace{\begin{bmatrix} \mathbf{0} \\ -\nabla \tilde{C}(\boldsymbol{\theta}) + \sigma \boldsymbol{M}^{1/2}\boldsymbol{R} \\ 0 \end{bmatrix} \mathrm{d}t}_{\mathrm{B}} +$$

$$\underbrace{\begin{bmatrix} \mathbf{0} \\ -\xi\boldsymbol{p}\mathrm{d}t + \sigma_{\mathrm{A}}\boldsymbol{M}^{1/2}\,\mathrm{d}\,\boldsymbol{W}_{\mathrm{A}} \\ 0 \end{bmatrix}}_{\mathrm{O}} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mu^{-1}\left[\boldsymbol{p}^{T}\boldsymbol{M}^{-1}\boldsymbol{p} - N_d\beta^{-1}\right] \end{bmatrix} \mathrm{d}t}_{\mathrm{D}}$$

- Here, we use the integration scheme: BADODAB.

Why Do We Care About β?

- Exponentially many local minima *[You and Wu, 2021]*
- Barren Plateau Phenomenon *[Boixo et al., 2018]*



The Effect of Temperature β^{-1}

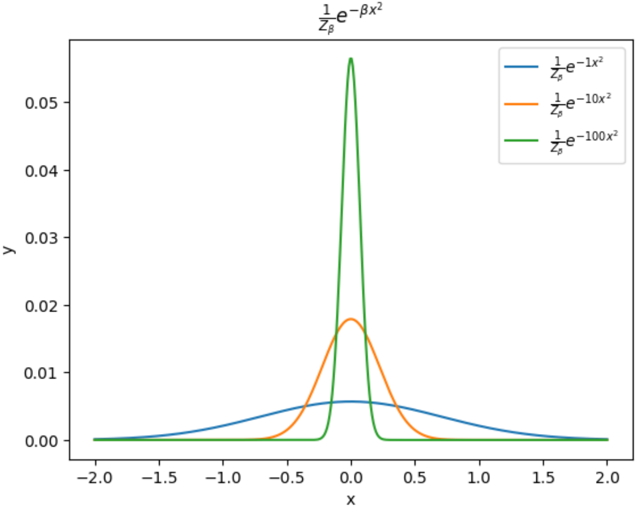
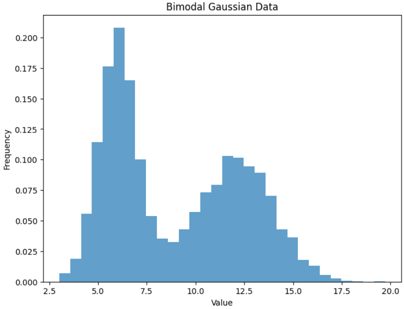


Figure 8: Effect of Temperature β^{-1}.

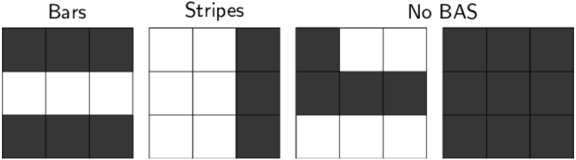
Results

Datasets

- **Bimodal Gaussian** (1-dimensional) [Samples = 5000]



- **Bars-and-Stripes** (9-dimensional) *[Burt et al, 1992]*



Hyperparameters

- Stepsize $h = 0.1$
- Reciprocal Temperature β
- Additive Noise $\sigma_A = 1$
- Batch Size $|S| = 200$
- Number of Samples (Bimodal) $= 5000$
- Number of Steps $= 5000$
- Number of Qubits $= 8$
- Circuit Depth $= 3$
- Thermal Mass $\mu = 1$
- Mass Matrix $= I$
- Cost Function $C(\theta) =$ MMD
- Prior $p(\theta) = 1$ (Uniform)

Bimodal Gaussian

Decreasing the temperature (Increasing β)

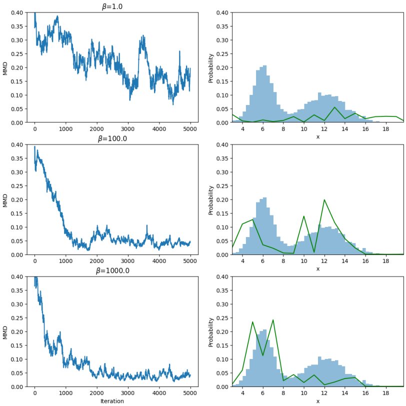


Figure 9: Increasing β yields faster mixing.

Proof of Concept

- Assume $\tilde{C}(\boldsymbol{\theta}) = C(\boldsymbol{\theta}) + \alpha\boldsymbol{\varepsilon}$.
- α is multiplicative factor and $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \boldsymbol{I})$.
- Gaussian Gradient Noise Model with Identity covariance which satisfies the assumptions of **SGAdLT**.

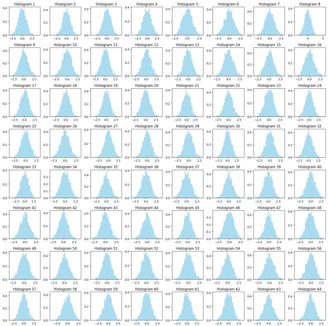
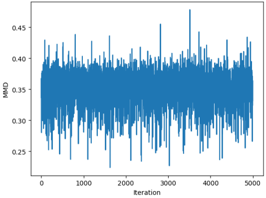
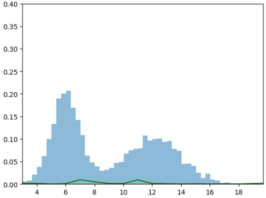


Figure 10: Gradient Noise Model $(\alpha = 1)$

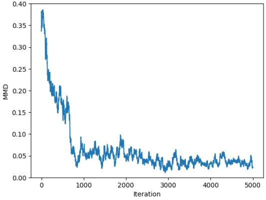
The effect of adding 10ε to $\nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta})$ i.e., $\alpha = 10$



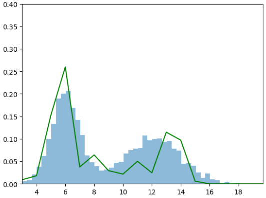
(a) Cost Curve for **SGLD**



(b) $p(y|\boldsymbol{D}) \approx \frac{1}{N}\Sigma_{i=1}^{N} p(y|\boldsymbol{\theta}_i^{SGLD})$



(c) Cost Curve for **SGAdLT**



(d) $p(y|\boldsymbol{D}) \approx \frac{1}{N}\Sigma_{i=1}^{N} p(y|\boldsymbol{\theta}_i^{SGAdLT})$

Figure 11: $\alpha = 10, \beta = 100$ (High Gradient Noise Regime)

Gradient Noise Model for $\tilde{C}(\boldsymbol{\theta})$

Recall our GNM for stochastic gradients in *Fig 12.* It may be Gaussian but definitely does not have constant covariance.

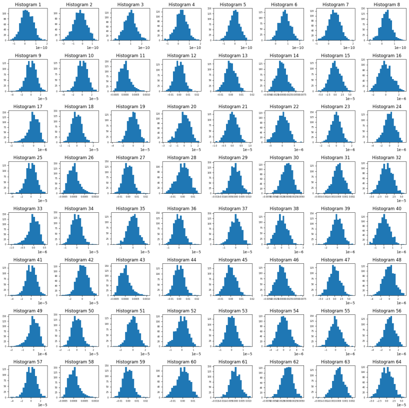
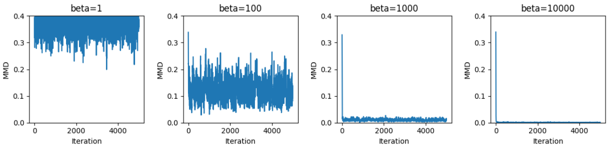
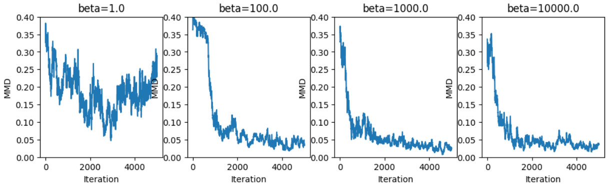


Figure 12: Component-wise Distributions of bias (Batch Size = 200)

SGLD vs SGAdLT on β (MMD)

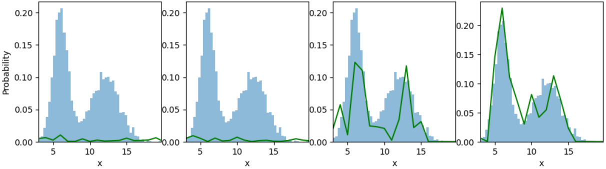


(a) Cost Curves for SGLD

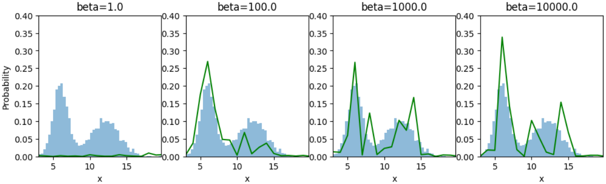


(b) Cost Curves for SGAdLT

Figure 13: Cost Curves with β varied. Notice how for higher β, SGLD acts as an optimizer instead of a sampler.



(a) Predictive Distribution $p(y|\boldsymbol{D})$ for SGLD



(b) Predictive Distribution $p(y|\boldsymbol{D})$ for SGAdLT

Future Steps (Hyperparameter Tuning)

- Grid Search to optimize hyperparameters:
  - Thermal Mass μ
  - Additive Noise σ_A
  - Mass Matrix M
  - Batch Size S
  - Reciprocal Temperature β
  - Number of Qubits
  - Laplace Prior to enforce sparsity
  - Using different kernels $k(.,.)$
- According to *[Sekkat and Stoltz, 2023]*, a modified Adaptive Langevin Thermostat may work well even for non-Gaussian (non-constant covariance) GNMs.

Bars and Stripes

Future Steps (BARS)

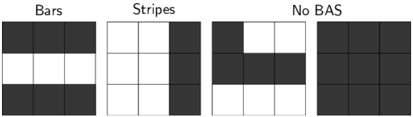


Figure 14: Bars-and-Stripes

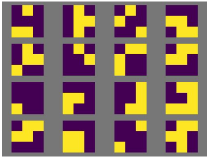


Figure 15: Progress so far

Thanks!