‘Such that’ problems turn up all over the place in ML

IMAGE GENERATION

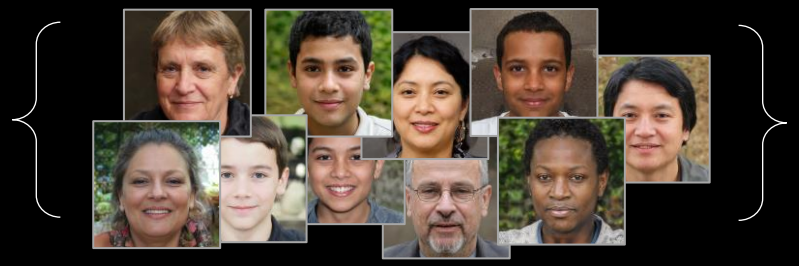
“Generate images,
such that they are indistinguishable from the real ones.”

LLM ALIGNMENT

“Maximize how highly your LLM’s replies are rated,
such that they’re not too far from normal text.”

We’ll see how “such that” is expressed and used in probabilistic ML.

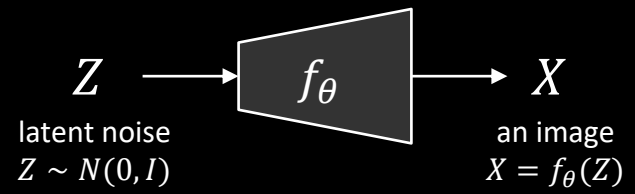
A generative modelling task



Training data: a dataset of photographs of real faces

Task: generate *new* images that look like them

Proposed model: *called a latent-variable generative model*



How should we train this model, to make $\Pr_{f_\theta(Z)}$ a good match for the true distribution of pictures? How should we evaluate it?

Fitting: choose θ to maximize $\log \Pr_{f_\theta(Z)}(\text{training photos})$

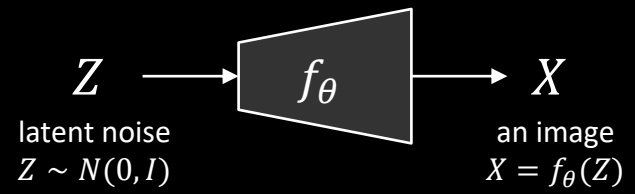
or, minimize $\mathrm{KL}(\Pr^{\text{train}} \parallel \Pr_{f_\theta(Z)})$

Evaluation: compare models by $\log \Pr_{f_\theta(Z)}(\text{holdout photos})$

or, $\mathrm{KL}(\Pr^{\text{holdout}} \parallel \Pr_{f_\theta(Z)})$

FACE IMAGES FROM THISPERSONNOTEXIST.ORG

Proposed model:



$$\Pr_{f_\theta(Z)}(x) = \int_z 1_{f_\theta(Z)=x} \Pr_Z(z)\, dz = \mathbb{E} 1_{f_\theta(Z)=x}$$

We can't solve this integral, and Monte Carlo is useless.
Should we give up on KL?

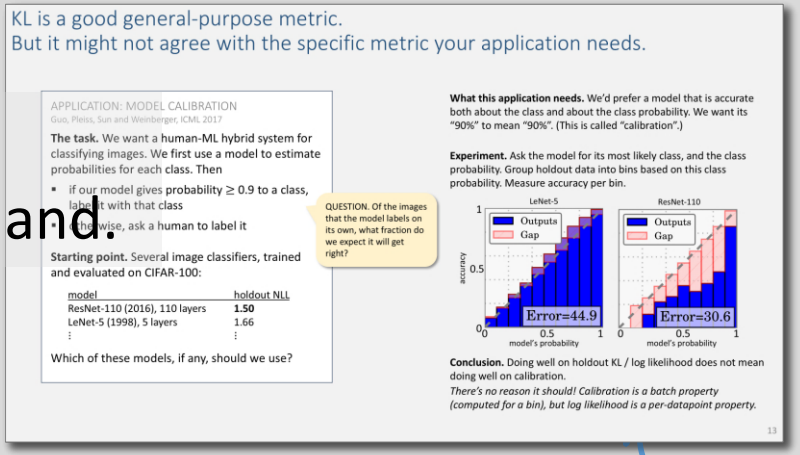
Fitting: choose θ to maximize $\log \Pr_{f_\theta(Z)}(\text{training photos})$

or, minimize $\mathrm{KL}(\Pr^{\text{train}} \parallel \Pr_{f_\theta(Z)})$

Evaluation: compare models by $\log \Pr_{f_\theta(Z)}(\text{holdout photos})$

or, $\mathrm{KL}(\Pr^{\text{holdout}} \parallel \Pr_{f_\theta(Z)})$

Last week, we discussed using specific metrics for the task at hand.



Incidentally, both of these are aggregate metrics (i.e. computed on a set, not per datapoint) so not friendly for training

Fréchet Inception Distance is widely used for measuring the quality of an image generator.

1. Take n real images.
   Generate n images using the generator.
2. Pass all $2n$ images through an ImageNet classifier (Inception-v3);
   keep 2048 of its internal features.
   giving a vector $f_i \in \mathbb{R}^{2048}$ for each real image, and $g_i \in \mathbb{R}^{2048}$ for each generated image
3. Summarize each bag of images by the mean and covariance of its features.
   model them as $F \sim N(\mu_F, \Sigma_F)$ and $G \sim N(\mu_G, \Sigma_G)$
4. Report the distance between the two summary distributions. Smaller is better.
   the distance is $\mathbb{E}\|F - G\|^2$ for the coupling that makes this smallest

Here's another statistic, one that isn't limited to images

$R =$ 

$S_\theta =$ 

Consider two bags of items:
real datapoints $R = \{x_1, \ldots, x_n\}$ and synthetic $S_\theta = \{f_\theta(z_1), \ldots, f_\theta(z_n)\}$

Task: tune θ so the two bags can't be told apart

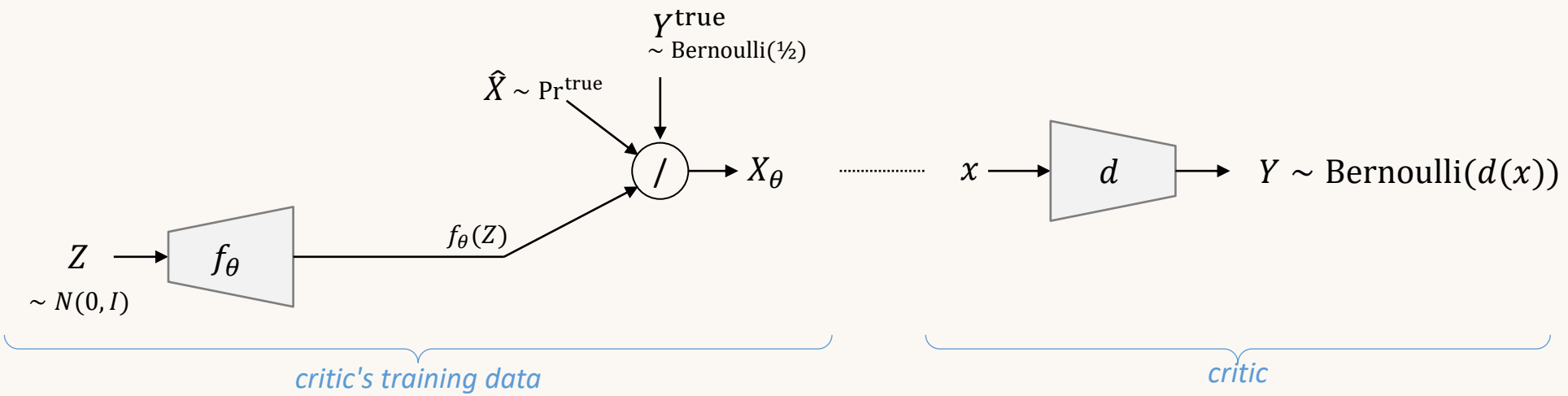
Cunning idea: measure how different the bags are, by how well a critic can tell them apart.

Proposal.

- The critic's task: given an item drawn at random from $R \cup S_\theta$, guess which bag it came from
- For a given θ, train the best critic we can, and measure its accuracy
- Tune θ to make this best-critic's-accuracy as small as possible

1. If the generator is great and the two bags are indistinguishable, what is the best-critic's accuracy?
2. If the generator is poor, what's the accuracy?

The maths of critic-based training + evaluation (called GAN)



A random item drawn from the two bags:
consider the random pair $(X_\theta, Y^{\text{true}})$ generated by first picking $Y^{\text{true}} \sim \text{Bernoulli}(½)$, then

$$X_\theta = \begin{cases} \text{a real image } \hat{X} & \text{if } Y^{\text{true}} = 1 \\ f_\theta(Z) & \text{if } Y^{\text{true}} = 0 \end{cases}$$

The critic: let $x \mapsto d(x) \in [0,1]$ be a probabilistic classifier, guessing how likely it is that item x is real.

The critic's score on an item (x, y^{true}) is

$$\log \Pr_Y(y^{\text{true}}|x) = 1_{y^{\text{true}}=1} \log d(x) + 1_{y^{\text{true}}=0} \log(1 - d(x))$$

so the average score on a random item is

$$L(f_\theta, d) := \mathbb{E}_{(X_\theta, Y^{\text{true}})} \log \Pr_Y(Y^{\text{true}}|X_\theta)$$
$$= \frac{1}{2}\mathbb{E} \log d(\hat{X}) + \frac{1}{2}\mathbb{E} \log\big(1 - d(f_\theta(Z))\big)$$

For this probabilistic-type critic:
If f_θ is perfect (and so the two bags are indistinguishable), what is the best critic $\hat{d}$, and what is $L(f_\theta, \hat{d})$?

The best critic. Given a family of critics d_ϕ, e.g. a neural network with weights ϕ, the best-scoring critic solves $\max_\phi L(f_\theta, d_\phi)$.

The best generator. Tune θ so that the best critic is as weak as possible: $\min_\theta \max_\phi L(f_\theta, d_\phi)$

Consider two distributions: the real distribution $q \coloneqq \Pr^{\text{true}}$ and our model's distribution $p_\theta \coloneqq \Pr_{f_\theta(Z)}$

Task: tune θ to make $p_\theta \approx q$

THE GAN'S GOAL

Let $L(f, d)$ be the success of critic d at distinguishing generator f. The GAN seeks

$$\min_\theta \max_\phi L(f_\theta, d_\phi)$$

For a perfect critic, the objective is

$$\min_\theta \{\mathrm{KL}(q \parallel m_\theta) + \mathrm{KL}(p_\theta \parallel m_\theta) + \text{const}\}$$

Jensen–Shannon divergence

where $m_\theta(x) \coloneqq {}^1\!/_2 \left(q(x) + p_\theta(x)\right)$

THE KL GOAL FROM LAST WEEK

$$\min_\theta \mathrm{KL}(q \parallel p_\theta)$$

The GAN seeks to minimize the JS divergence, but without actually needing to evaluate $p_\theta(x)$.
Can we pull off the same trick for KL divergence?

There is a magic trick that's helpful here: introducing a spurious 'such that' constraint.

EXERCISE

Consider two distributions q and p_θ on outcomes $\{1,2,3\}$, given by

x	1	2	3
q_x	0.5	0.3	0.2
$p_\theta(x)$	θ	θ	$1-2\theta$

The goal: minimize $\mathrm{KL}(q \parallel p_\theta) = \sum_x q_x \log {}^{q_x}/_{p_\theta(x)}$ over θ

The goal. minimize $\mathrm{KL}'(q \parallel p_\theta)$ over θ where $\mathrm{KL}'(q \parallel p_\theta)$ is the solution to

$$\text{minimize } \sum_x q_x \log {}^{q_x}/_{r_x} \text{ over } r \in \mathbb{R}^3_{>0} \text{ such that } r_x = p_\theta(x)\ \forall x$$

The Lagrangian for the inner opt., with a dual variable λ_x on each constraint:

$$\mathcal{L}(\theta, r; \lambda) := \sum_x q_x \log \frac{q_x}{r_x} + \sum_x \lambda_x (r_x - p_\theta(x))$$

1. For which $\lambda \in \mathbb{R}^3$ does $\mathcal{L}(\theta, r; \lambda)$ have a finite minimum over r?
2. For such λ, find the minimizing r and the minimum value.
3. Find the λ that makes this minimum as large as possible.
   (For this, to be concrete, use $\theta = 0.3$.)
4. For given θ and λ, what is the relationship between $\mathrm{KL}(q \parallel p_\theta)$ and $\min_r \mathcal{L}(\theta, r; \lambda)$?

The maths of KL minimization, GAN-style

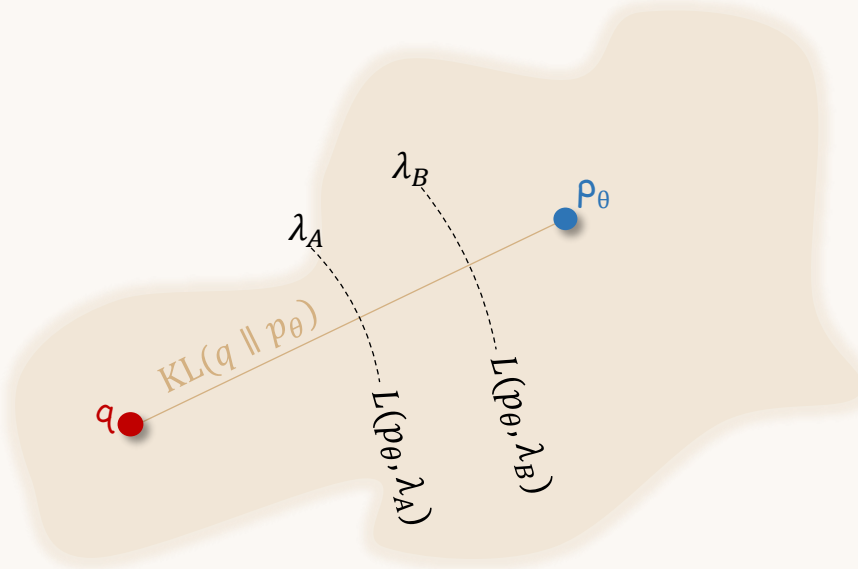
The goal. minimize $\mathrm{KL}'(q \parallel p_\theta)$ over θ where $\mathrm{KL}'(q \parallel p_\theta)$ is the solution to

$$\text{minimize } \textstyle\sum_x q_x \log {}^{q_x}/_{r_x} \text{ over } r > 0 \text{ such that } r_x = p_\theta(x)\ \forall x$$

notation: r>0 means componentwise, i.e. r_x>0 ∀x

The Lagrangian for the inner optimization is

$$\mathcal{L}(\theta, r; \lambda) \coloneqq \textstyle\sum_x q_x \log {}^{q_x}/_{r_x} + \sum_x \lambda_x (r_x - p_\theta(x))$$



Weak duality. For any θ and any $\lambda > 0$,

$$\begin{aligned}
\mathrm{KL}(q \parallel p_\theta) &= \mathcal{L}(\theta, p_\theta; \lambda) && \text{since when r is right the penalty disappears} \\
&\geq \min_{r>0} \mathcal{L}(\theta, r; \lambda) && \text{generic lower-bound maths} \\
&= \Sigma_x q_x \log \lambda_x - \Sigma_x \lambda_x p_\theta(x) + 1 && \text{same algebra as in the exercise} \\
&=: L(p_\theta, \lambda) && \text{Any critic λ tells me "Your model is at least this far away from the truth"}
\end{aligned}$$

Fighting the harshest critic. Given a family of critics $x \mapsto \lambda_\phi(x) > 0$,

e.g. a neural network with weights ϕ,

$$\mathrm{KL}(q \parallel p_\theta) \geq \max_\phi L(p_\theta, \lambda_\phi)$$

my harshest critic gives the most useful lower bound

thus

$$\min_\theta \mathrm{KL}(q \parallel p_\theta) \geq \min_\theta \max_\phi L(p_\theta, \lambda_\phi).$$

Design my generator to push this harshest-critic bound as low as possible

Our objective is easy to estimate by sampling.

$$\begin{aligned}
L(p_\theta, \lambda_\phi) &= \sum_x q_x \log \lambda_\phi(x) - \sum_x \lambda_\phi(x)\, p_\theta(x) + 1 \\
&= \mathbb{E}_{X \sim q} \log \lambda_\phi(X) \; - \; \mathbb{E}_{X \sim p_\theta} \lambda_\phi(X) + 1
\end{aligned}$$

average over real photos

average over generated photos

This minimax formula is given more generally by Nowozin, Cseke and Tomioka 2016 (f-GAN), with a slick proof for continuous spaces adapted from Nguyen, Wainwright and Jordan 2010.

There is a common thread through all of this: the density ratio $q_x/p_\theta(x)$.

GAN CRITIC

In the GAN formulation, what task was the critic trying to solve?

The critic classifies an image as real or synthetic.
The best-scoring critic matches the true frequency:

$$\hat{d}(x) = \mathbb{P}(\text{real} \mid x) = \frac{q_x}{q_x + p_\theta(x)}$$

This is just a transformed version of $\hat{\lambda}$:

$$\hat{d}(x) = \frac{\hat{\lambda}_x}{1 + \hat{\lambda}_x}$$

BAYESIAN CLASSIFICATION

What is the probability that a given image x is real, i.e. $\mathbb{P}(\text{real}|x)$?

I generate an image X as follows: either real ($X \sim q$) with probability π, or synthetic ($X \sim p_\theta$) with probability $1 - \pi$. Then

$$\mathbb{P}(\text{real} \mid x) = \frac{\pi q_x}{\pi q_x + (1-\pi)p_\theta(x)} = \frac{\hat{\lambda}_x}{\hat{\lambda}_x + (1-\pi)/\pi}$$

LAGRANGIAN CRITIC

The standard Lagrangian procedure for solving "minimize $f(r)$ over r such that $g(r) = 0$":

1. Lagrangian $\mathcal{L}(r;\lambda) = f(r) + \lambda g(r)$
2. For what λ is there a finite $\min_r$?
3. For given λ find the minimizing $\hat{r}$
4. ???

Every critic λ gives a lower bound on the KL:

$$\mathrm{KL}(q \parallel p_\theta) \geq \min_{r>0} \mathcal{L}(\theta, r; \lambda)$$

The harshest critic gives the tightest bound. It is

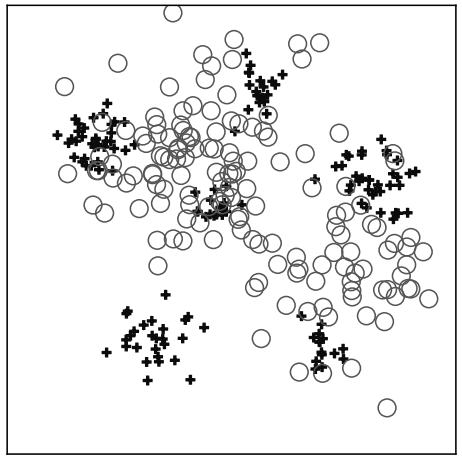
$$\hat{\lambda}_x = \frac{q_x}{p_\theta(x)}$$

Intuition: $\hat{\lambda}_x$ is looking for places where the model is a bad match for the truth. The equation for the lower bound

$$\mathrm{KL}(q \parallel p_\theta) \geq \mathbb{E}_{X\sim q} \log \lambda_\phi(X) \; - \; \mathbb{E}_{X\sim p_\theta} \lambda_\phi(X) + 1$$

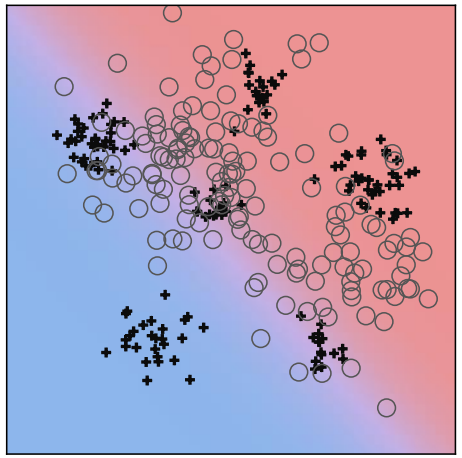
shows how it works:
it puts more weight where $q_x > p_\theta(x)$, less where $q_x < p_\theta(x)$.

Why bother training a critic network at all? We have a formula for $\hat{\lambda}$, so why not just use it?

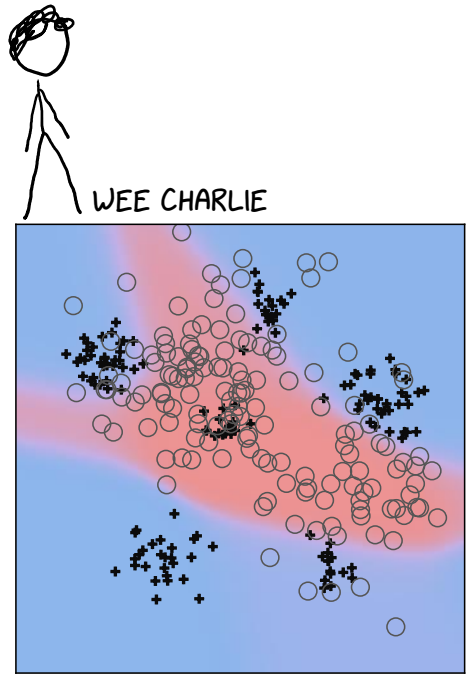


What do different critics pick out?

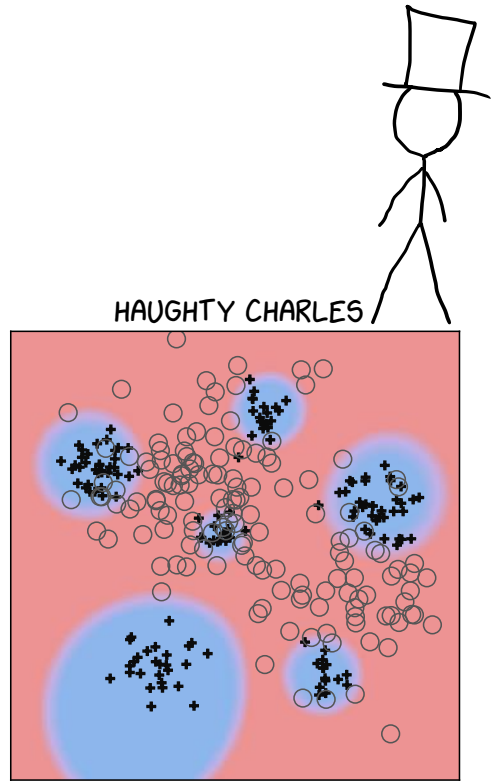
+ real data
○ generated *(this generator is eight equal-weight Gaussian blobs, free to change location and scale)*



very weak critic
$\lambda_{\mathrm{vwc}}(x) = e^{\alpha+\beta\cdot x}$

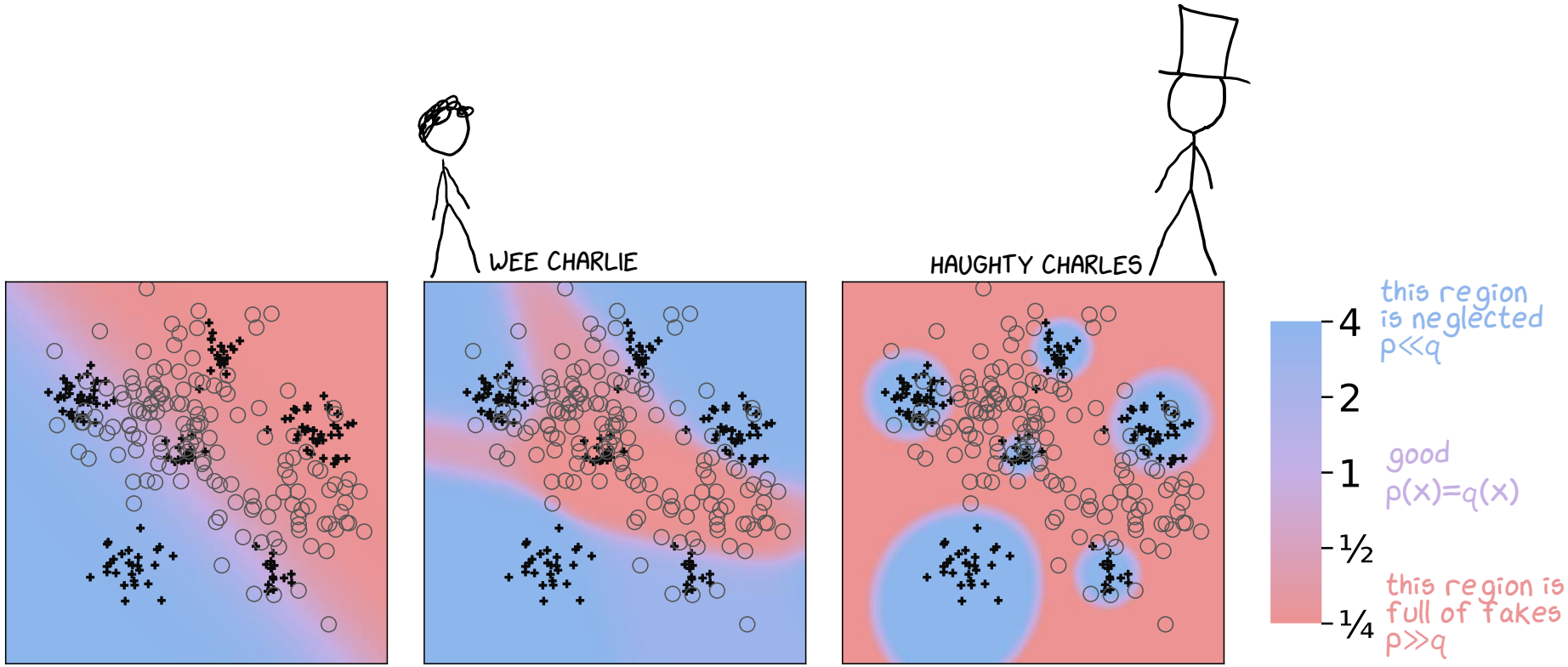


weak critic
$\lambda_{\mathrm{wc}} = 4\text{-unit network}$



harshest critic
$\lambda_{\mathrm{hc}}(x) = q(x)/p(x)$

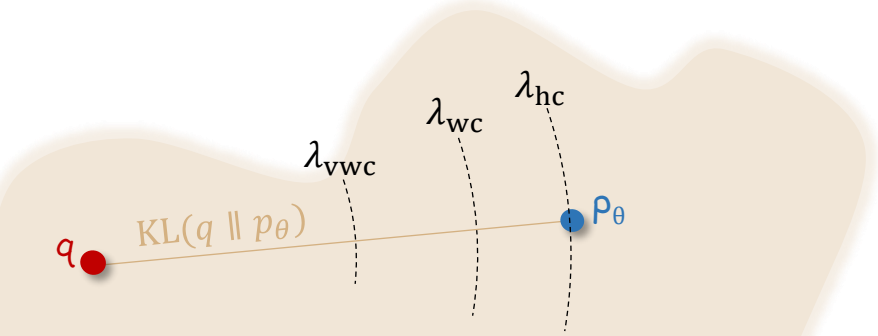




very weak critic
$\lambda_{\text{vwc}}(x) = e^{\alpha+\beta\cdot x}$

weak critic
λ_{wc} = 4-unit network

harshest critic
$\lambda_{\text{hc}}(x) = q(x)/p(x)$



Every critic λ gives a lower bound on the KL:

$$\text{KL}(q \parallel p_\theta) \geq \mathbb{E}_{X\sim q}\log\lambda(X) \; - \; \mathbb{E}_{X\sim p_\theta}\lambda(X) + 1$$

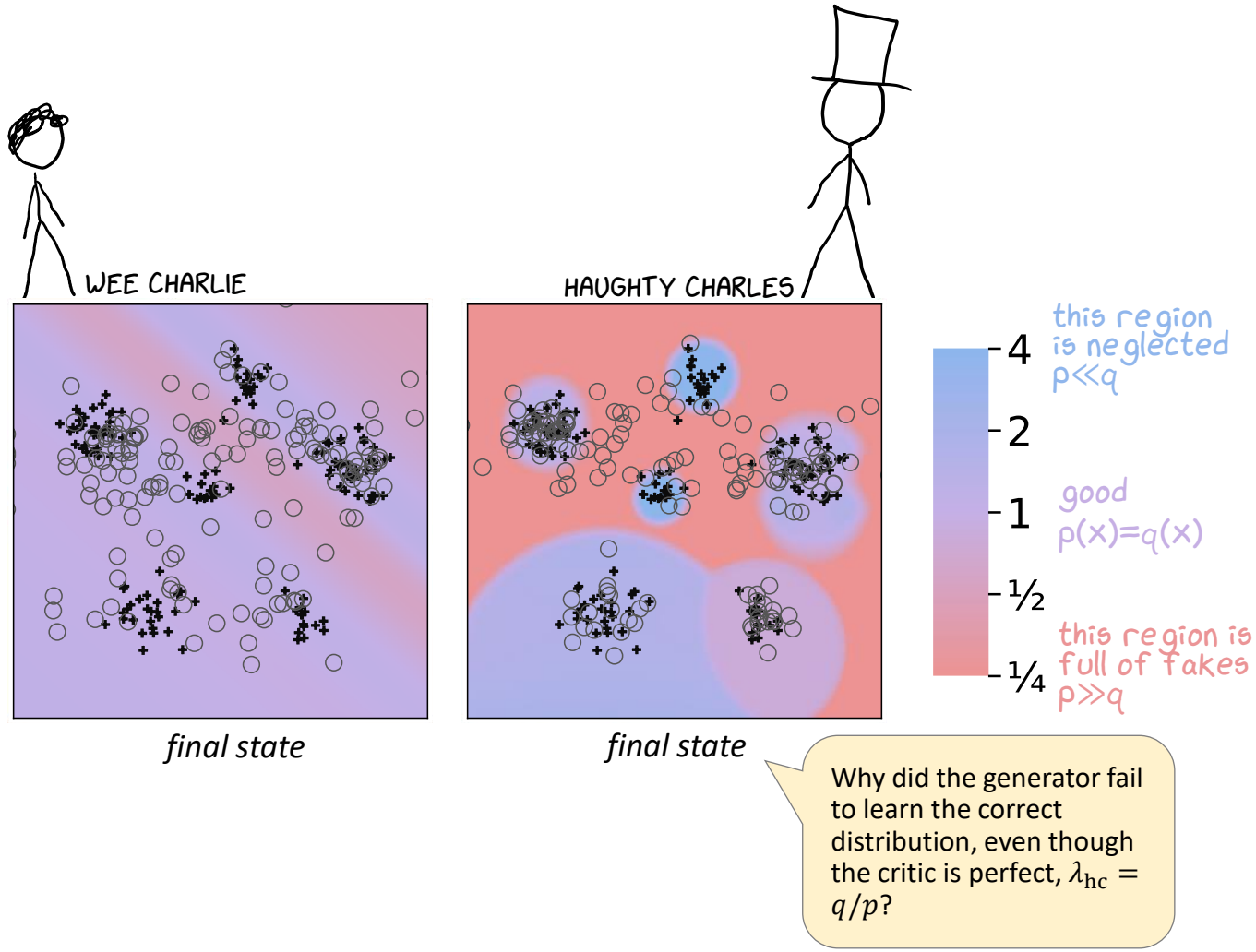
The harshest critic λ_{hc} gives an exact bound, since

$$\mathbb{E}_{X\sim q}\log\lambda_{\text{hc}}(X) - \mathbb{E}_{X\sim p}\lambda_{\text{hc}}(X) + 1 = \mathbb{E}_{X\sim q}\log\frac{q(X)}{p(X)} - \int_x \frac{q(x)}{p(x)}p(x)\,dx + 1 = \text{KL}(q \parallel p)$$

As we train the generator, the critic and the generator play a cat-and-mouse game.

- The critic paints regions red when there are too many generated samples, blue when too few
- The generator scurries into blue areas

Once the critic can no longer classify successfully, there is no more improvement.



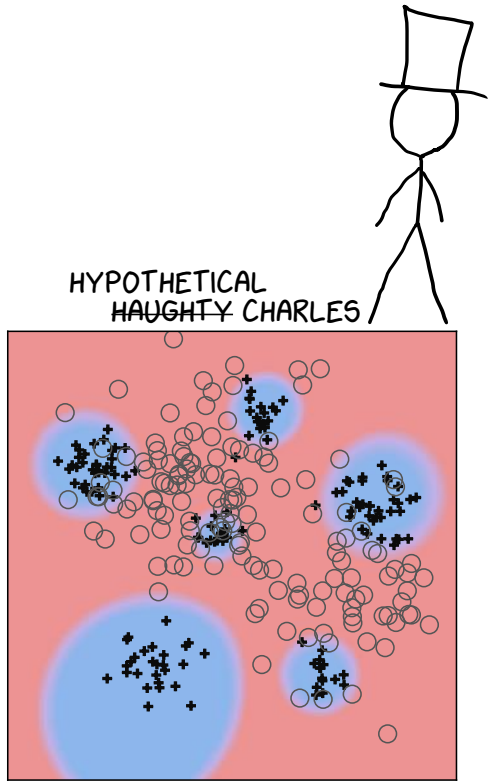
In these toy examples, the harshest critic λ_{hc} wasn't trained. This is a toy example, we know q and p, and we simply defined $\lambda_{\text{hc}} = q/p$.

Suppose instead we naively train a very high-capacity critic on the training dataset:

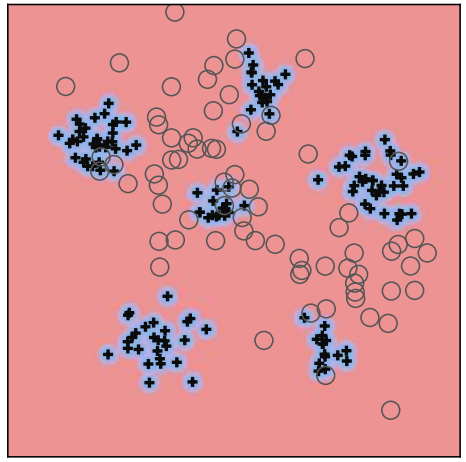
$$\max_\lambda \left\{ \mathbb{E}_{X\sim\hat{q}^{\text{train}}} \log \lambda(X) - \mathbb{E}_{X\sim p}\lambda(X) + 1 \right\}$$

What will this high-capacity critic learn?

When we train the critic we need to regularize it, as usual, so that it penalizes "deviation from q" rather than "deviation from $\hat{q}^{\text{train}}$".



harshest critic
$\lambda_{\text{hc}}(x) = q(x)/p(x)$



overfitted critic

The critic needs to be weak, to teach the generator how to improve

Realistic images are a tiny thin manifold in the space of all images



$$p(x_0) \leftarrow p(x_0) - \delta$$
$$p(x_1) \leftarrow p(x_1) + \delta$$

Suppose the true distribution q has zero mass outside a small manifold

Let's use the perfect harsh critic $\lambda_{\text{hc}}(x) = q(x)/p(x)$ *i.e. we'll assume away the problem of overfitting!*

The generator adapts p to minimize

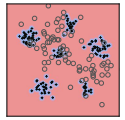
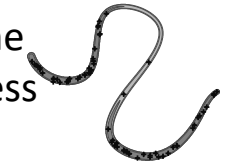
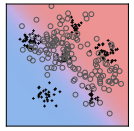
$$\mathbb{E}_{X\sim q} \log \lambda_{\text{hc}}(X) - \underbrace{\mathbb{E}_{X\sim p} \lambda_{\text{hc}}(X)}_{\text{the bit we want to optimize}} + 1$$

It does this by gradient descent,
i.e. by looking at small changes in p and asking how they improve the objective.

Consider a small change in p to make it more like q:
move a bit of probability mass from x_0 to a nearby x_1.

1. Does this improve $\mathbb{E}_{X\sim p} \lambda_{\text{hc}}(X)$?
2. How could we fix this problem? I.e. how can we ensure there's a gradient pushing p towards q?

How is the field coping with the challenges of training GANs?

challenge	the fix	in the field
it's all too finicky 😭	give up and **try something else** :-)	diffusion models, flow matching
overfitted critic 	**regularize the critic**	gradient penalties, spectral norm, early stopping; "the modern GAN baseline"
lost in the wilderness 	**weaken the critic** (or **fuzz the truth**: add noise to blur the manifold)	instance noise; Diffusion-GAN; critics at every noise level of a diffusion model
weak critic is blind to what matters 	**anchor the generator**: initialize the generator to something decent, and only use the critic to sharpen it	distilling diffusion into one-step generators

Aside from training generators, the general idea of the *critic* comes up again and again,
as a component in other pipelines, whenever we need a "such that" flavour. As we now see ...