Table 2: Notations for GPP.

Terminology	Symbol	Meaning
Model	$\mathcal{M}$	The model to probe.
Stimuli space	$\mathcal{X}$	The space of stimuli.
Representation space	$\mathcal{A} \subseteq \mathbb{R}^d$	The space of vector representations of $\mathcal{M}$.
Basis functions	ϕ	The function (contained in $\mathcal{M}$) mapping from $\mathcal{X}$ to $\mathcal{A}$.
Vector representation	$\phi(x)$	The vector representation of a stimulus $x \in \mathcal{X}$.
	α	$\alpha : \mathcal{A} \mapsto \mathbb{R}^+$, mapping to the 1st parameter of a Beta distribution.
	β	$\beta : \mathcal{A} \mapsto \mathbb{R}^+$, mapping to the 2nd parameter of a Beta distribution.
	t_α	A random function such that $t_\alpha(a) \sim \text{Gamma}(\alpha(a), 1)$.
	t_β	A random function such that $t_\beta(a) \sim \text{Gamma}(\beta(a), 1)$.
Beta distribution	$\text{Beta}(\alpha(a), \beta(a))$	The Beta distribution for $g(a)$, $\forall a \in \mathcal{A}$.
Mean function	μ	$\mu : \mathcal{A} \mapsto \mathbb{R}$.
Kernel function	k	$k : \mathcal{A} \times \mathcal{A} \mapsto \mathbb{R}$.
GP	$\mathcal{GP}(\mu, k)$	A GP with mean function μ and kernel function k.
	$f_\alpha \sim \mathcal{GP}(\mu, k)$	$f_\alpha : \mathcal{A} \mapsto \mathbb{R}$, the (latent) function for approximating t_α.
	$f_\beta \sim \mathcal{GP}(\mu, k)$	$f_\beta : \mathcal{A} \mapsto \mathbb{R}$, the (latent) function for approximating t_β.
	θ	Parameter for the Beta GP in GPP, $\theta = (\mu, k)$.
Beta GP	$\mathcal{G}(\theta)$	A distribution over functions mapping from $\mathcal{A}$ to $[0, 1]$.
Classifier	$g \sim \mathcal{G}(\theta)$	$g = \frac{1}{1+e^{-f}} : \mathcal{A} \mapsto [0, 1]$, a random function.
Observations	D	$D = \{(\phi(x_i), y_i)\}_{i=1}^{\lvert D \rvert}$, $x_i \in \mathcal{X}, y_i \in \{0, 1\}$.
Beta GP posterior	$\mathcal{G}(\theta \mid D)$	The Beta GP conditional on observations D.
Queries	$\boldsymbol{a}_q$	$\boldsymbol{a}_q = [\phi(x_j)]_{j=1}^{\lvert \boldsymbol{a}_q \rvert} \in \mathbb{R}^{d \times \lvert \boldsymbol{a}_q \rvert}$.
Predicted probabilities	$g(\boldsymbol{a}_q)$	$g(\boldsymbol{a}_q) = [g(a)]_{a \in \boldsymbol{a}_q}$.

A Notation

In Table 2, we include the main notation used in this paper.

B Details on the Beta GP in GPP

As shown in § 2.2 and §2.3, with observations $D = \{(a_i, y_i)\}_{i=1}^{|D|}$, the posterior of a Beta GP can be written as the transformed version of a GP, i.e.,

$$g = \frac{1}{1 + e^{-f}} \sim \mathcal{G}(\theta \mid D), \text{ where } f \sim \mathcal{GP}(\mu_D, k_D).$$

We show how to obtain the mean and kernel functions μ_D, k_D in §B.1, as well as the posterior of weights in §B.2 for the cosine kernel. The behavior of GPP relies on two hyperparameters, ϵ and s, and we explain what they are and how to set them in §B.3. In §B.4, we analyze and bound the episteme of GPP.

As a reference, Figure 8 shows the graphical model of GPP.

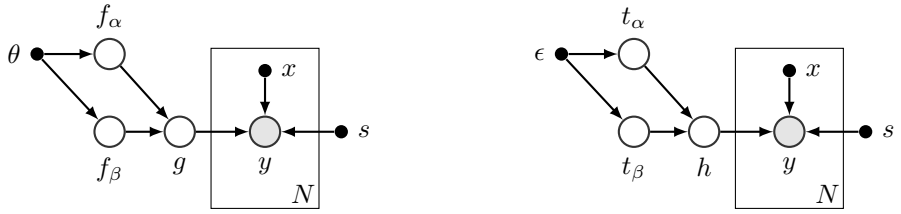

Figure 8: Left: Graphical model of the Beta GP in GPP with N observations. Right: The stochastic process that a Beta GP approximates, where $h = \frac{t_\alpha}{t_\alpha + t_\beta} \sim \text{Beta}(\epsilon, \epsilon), t_\alpha \sim \text{Gamma}(\epsilon, 1), t_\beta \sim \text{Gamma}(\epsilon, 1)$, and $t_\alpha \perp\!\!\!\perp t_\beta$. In Beta GP, $g = \frac{e^{f_\alpha}}{e^{f_\alpha} + e^{f_\beta}}$, where f_α approximates $\log t_\alpha$, and f_β approximates $\log t_\beta$. Hyperparameter s is the weight for each observation.

B.1 Posterior inference (extension of §2.3.2)

Without loss of generality, we write observations as a union of a dataset (of size n) with the positive labels only and a dataset (of size $|D| - n$) with negative labels only, i.e., $D = \{(a_i, y_i)\}_{i=1}^{|D|} = D^+ \cup D^-$ where $D^+ = \{(a_i, y_i)\}_{i=1}^{n}$ and $D^- = \{(a_i, y_i)\}_{i=n+1}^{|D|}$.

For convenience, we use the following short-hand notation:

$$v' = \log(\frac{1}{\epsilon + s} + 1), \quad v'' = \log(\frac{1}{\epsilon} + 1), \quad y' = \log(\epsilon + s) - \frac{v'}{2}, \quad y'' = \log(\epsilon) - \frac{v''}{2}.$$

Observing a positive example is equivalent to observing y' with noise $\mathcal{N}(0, v')$ on f_α and observing y'' with noise $\mathcal{N}(0, v'')$ on f_β. Vice versa for observing a negative example. We also denote 1_m as a column vector of size m filled with 1s, and I_m as an identity matrix of size m.

Recall that $f_\alpha \sim \mathcal{GP}(\mu, k), f_\beta \sim \mathcal{GP}(\mu, k)$ and $f_\alpha \perp\!\!\!\perp f_\beta$. The posterior for f_α is $f_\alpha \mid D \sim \mathcal{GP}(\mu_\alpha, k_\alpha)$. For any $a, a' \in \mathcal{A}$,

$$\mu_\alpha(a) = \mu(a) + k(a, \boldsymbol{a}) K_\alpha^{-1}(\boldsymbol{y}_\alpha - \mu(\boldsymbol{a})), \quad k_\alpha(a, a') = k(a, a') - k(a, \boldsymbol{a}) K_\alpha^{-1} k(\boldsymbol{a}, a'), \tag{5}$$

where

$$k(a, \boldsymbol{a}) = [k(a_i, a)]_{i=1}^{|D|} \in \mathbb{R}^{1 \times |D|}, \quad k(\boldsymbol{a}, a') = [k(a_i, a')]_{i=1}^{|D|} \in \mathbb{R}^{|D| \times 1},$$
$$\mu(\boldsymbol{a}) = [\mu(a_i)]_{i=1}^{|D|} \in \mathbb{R}^{|D| \times 1}, \quad K = [k(a_i, a_j)]_{i=1,j=1}^{|D|} \in \mathbb{R}^{|D| \times |D|},$$

and

$$\boldsymbol{y}_\alpha = \begin{bmatrix} y' 1_n \\ y'' 1_{|D|-n} \end{bmatrix} \in \mathbb{R}^{|D| \times 1}, \quad K_\alpha = K + \begin{bmatrix} v' I_n & 0 \\ 0 & v'' I_{|D|-n} \end{bmatrix} \in \mathbb{R}^{|D| \times |D|}.$$

Similarly for $f_\beta \mid D \sim \mathcal{GP}(\mu_\beta, k_\beta)$, we have

$$\mu_\beta(a) = \mu(a) + k(a, \boldsymbol{a}) K_\beta^{-1}(\boldsymbol{y}_\beta - \mu(\boldsymbol{a})), \quad k_\beta(a, a') = k(a, a') - k(a, \boldsymbol{a}) K_\beta^{-1} k(\boldsymbol{a}, a'), \tag{6}$$

where

$$\boldsymbol{y}_\beta = \begin{bmatrix} y'' 1_n \\ y' 1_{|D|-n} \end{bmatrix} \in \mathbb{R}^{|D| \times 1}, \quad K_\beta = K + \begin{bmatrix} v'' I_n & 0 \\ 0 & v' I_{|D|-n} \end{bmatrix} \in \mathbb{R}^{|D| \times |D|}.$$

Since $f = f_\alpha - f_\beta$, by combining Eq. 5 and Eq. 6, we have $f \mid D \sim \mathcal{GP}(\mu_D, k_D)$, and

$$\mu_D(a) = \mu_\alpha(a) - \mu_\beta(a) = k(a, \boldsymbol{a}) \left(K_\alpha^{-1}(\boldsymbol{y}_\alpha - \mu(\boldsymbol{a})) - K_\beta^{-1}(\boldsymbol{y}_\beta - \mu(\boldsymbol{a})) \right),$$
$$k_D(a, a') = k_\alpha(a, a') + k_\beta(a, a') = 2k(a, a') - k(a, \boldsymbol{a}) \left(K_\alpha^{-1} + K_\beta^{-1} \right) k(\boldsymbol{a}, a'). \tag{7}$$

Thus, we obtain the closed-form posterior for function f.

For classifier $g(a) = \frac{1}{1 + e^{-f(a)}}$, we can then get its PDF as follows,

$$p_{g(a)}(y) = \frac{1}{y(1-y)\sqrt{2\pi k_D(a, a)}} \exp\left(-\frac{(\log(y) - \log(1-y) - \mu_D(a))^2}{2k_D(a, a)} \right). \tag{8}$$

B.2 Posterior inference for weights (extension of §2.3.3)

If we use the cosine kernel in §2.3.3, the posterior of f_α can be written as

$$f_\alpha(a) = W_\alpha^\top \psi(a) + \mu(a), \text{ where } W_\alpha \mid D \sim \mathcal{N}(u_\alpha, \Sigma_\alpha), W_\alpha \in \mathbb{R}^{d+1}.$$

This means $f_\alpha(a) \mid D \sim \mathcal{N}(u_\alpha^\top \psi(a) + \mu(a), \psi(a)^\top \Sigma_\alpha \psi(a))$.

Because of Eq. 5 and Eq. 4, we can also write the posterior of f_α as

$$\begin{aligned}\mu_\alpha(a) &= \mu(a) + \psi(a)^\top \psi(\boldsymbol{a}) K_\alpha^{-1}(\boldsymbol{y}_\alpha - \mu(\boldsymbol{a})),\\ k_\alpha(a, a') &= \psi(a)^\top \psi(a) - \psi(a)^\top \psi(\boldsymbol{a}) K_\alpha^{-1} \psi(\boldsymbol{a})^\top \psi(a).\end{aligned}$$

By comparing the above two ways of writing the posterior of f_α, we obtain

$$u_\alpha = \psi(\boldsymbol{a}) K_\alpha^{-1}(\boldsymbol{y}_\alpha - \mu(\boldsymbol{a})), \quad \Sigma_\alpha = I_{d+1} - \psi(\boldsymbol{a}) K_\alpha^{-1} \psi(\boldsymbol{a})^\top.$$

Similarly, for $f_\beta(a) = W_\beta^\top \psi(a) + \mu(a), W_\beta \mid D \sim \mathcal{N}(u_\beta, \Sigma_\beta)$, we have

$$u_\beta = \psi(\boldsymbol{a}) K_\beta^{-1}(\boldsymbol{y}_\beta - \mu(\boldsymbol{a})), \quad \Sigma_\beta = I_{d+1} - \psi(\boldsymbol{a}) K_\beta^{-1} \psi(\boldsymbol{a})^\top.$$

Then, for $f = f_\alpha - f_\beta = W^\top \psi(a), W \mid D \sim \mathcal{N}(\mu, \Sigma)$, we have

$$u = \psi(\boldsymbol{a}) \left(K_\alpha^{-1}(\boldsymbol{y}_\alpha - \mu(\boldsymbol{a})) - K_\beta^{-1}(\boldsymbol{y}_\beta - \mu(\boldsymbol{a})) \right), \Sigma = 2I_{d+1} - \psi(\boldsymbol{a}) \left(K_\alpha^{-1} + K_\beta^{-1} \right) \psi(\boldsymbol{a})^\top.$$

This means we can directly sample classifiers from a Beta GP with a cosine kernel by sampling weights W from a multivariate Gaussian distribution defined above.

B.3 How to set hyperparameters

There are two hyperparameters in GPP with the cosine kernel: ϵ, which determines the prior, and s, which determines the posterior.

For any $a \in \mathcal{A}$, the prior on the probability that the label is positive is Beta(ϵ, ϵ). As noted in §2.3.1, $\epsilon < 1$ reflects a belief that $g(a)$ should be close to either 0 or 1; $\epsilon = 1$ gives a uniform distribution over $[0, 1]$; and $\epsilon > 1$ reflects a belief that $g(a)$ is centered at 0.5. In the Beta GP, the Beta prior is approximated as

$$p_{g(a)}(y) = \frac{1}{y(1-y)\sqrt{4\pi \log(\frac{1}{\epsilon} + 1)}} \exp\left(-\frac{(\log(y) - \log(1-y))^2}{4 \log(\frac{1}{\epsilon} + 1)} \right). \tag{9}$$

Eq. 9 is obtained using the prior of f, i.e., $\mu_D(a) = 0, k_D(a, a) = 2k(a, a) = 2\log(\frac{1}{\epsilon} + 1)$, in Eq. 8. Users can choose Beta$(\hat{\epsilon}, \hat{\epsilon})$ for moment matching in Eq. 9 to get a better approximation of Beta(ϵ, ϵ).

Figure 9 shows both the PDF of Beta priors and the approximations.

For setting the hyperparameter s, users can also directly inspect the behaviors of different values of s and choose an appropriate value. Figure 10 and Figure 11 show how the posterior changes with one negative or two opposite-label observations. Larger s leads to more concentrated posterior.

Since all of these distributions are easily computable and can be visualized clearly, users can directly inspect the behaviors of these different hyperparameters and choose a suitable option.

B.4 Analyses of episteme (extension of §2.4)

For each $a \in \mathcal{A}$, episteme is the negative of $\mathbb{H}[g(a)]$. By Eq. 8, we have

$$\begin{aligned}\mathbb{H}[g(a)] &= -\int p_{g(a)}(y) \log p_{g(a)}(y) \, \mathrm{d}y\\ &= -\mathbb{E}[\log p_{g(a)}(y)]\\ &= \mathbb{E}[\log(y(1-y))] + \mathbb{H}[f(a)]\\ &< \mathbb{H}[f(a)].\end{aligned}$$

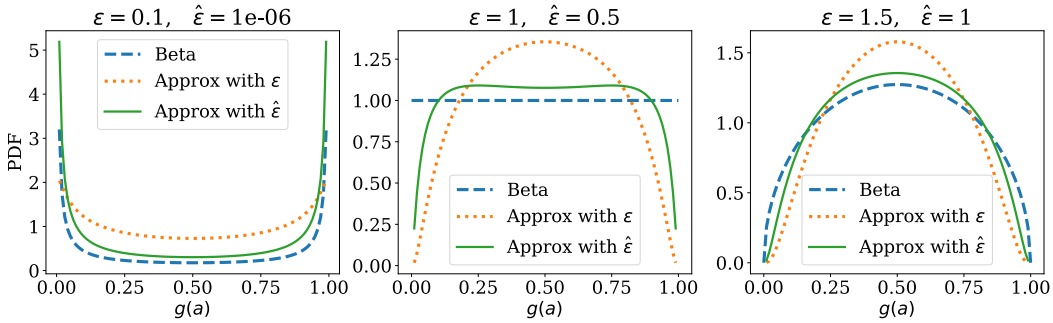

Figure 9: PDF of Beta(ϵ, ϵ) and the approximations that either use ϵ for moment matching or $\hat{\epsilon}$. Because both the PDF of Beta distributions and the approximations in Eq. 9 are easily computable, users can inspect the distributions directly and choose the right $\hat{\epsilon}$ to match with their own beliefs.

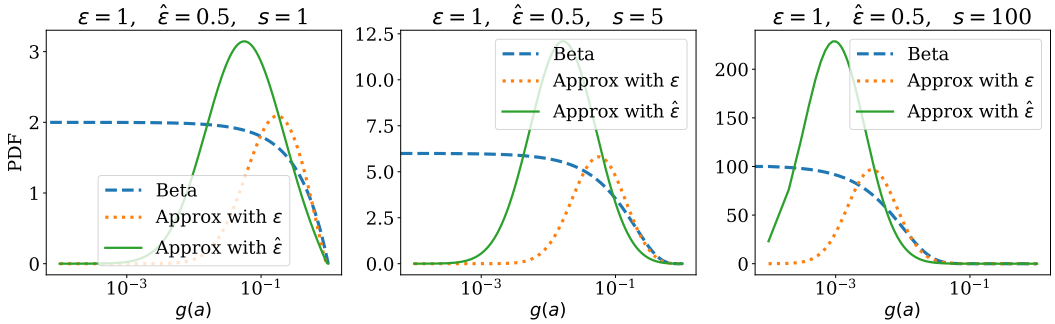

Figure 10: PDF of Beta$(\epsilon, \epsilon + s)$ and the approximates that either uses ϵ for moment matching or $\hat{\epsilon}$. These distributions are the (approximated) posteriors of $g(a)$ after observing 1 negative example. Hyperparameter s are 1 (Left), 5 (Middle) or 100 (Right), and with a larger s, the approximate becomes more concentrated at a lower value of $g(a)$.

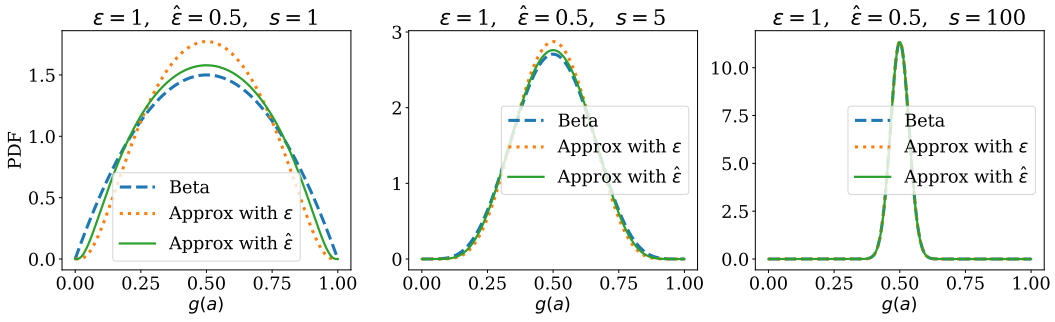

Figure 11: The same setup as Figure 10, except that the observations include 1 positive and 1 negative examples at the same representation a. A larger s results in a PDF that is more concentrated at $g(a) = 0.5$.

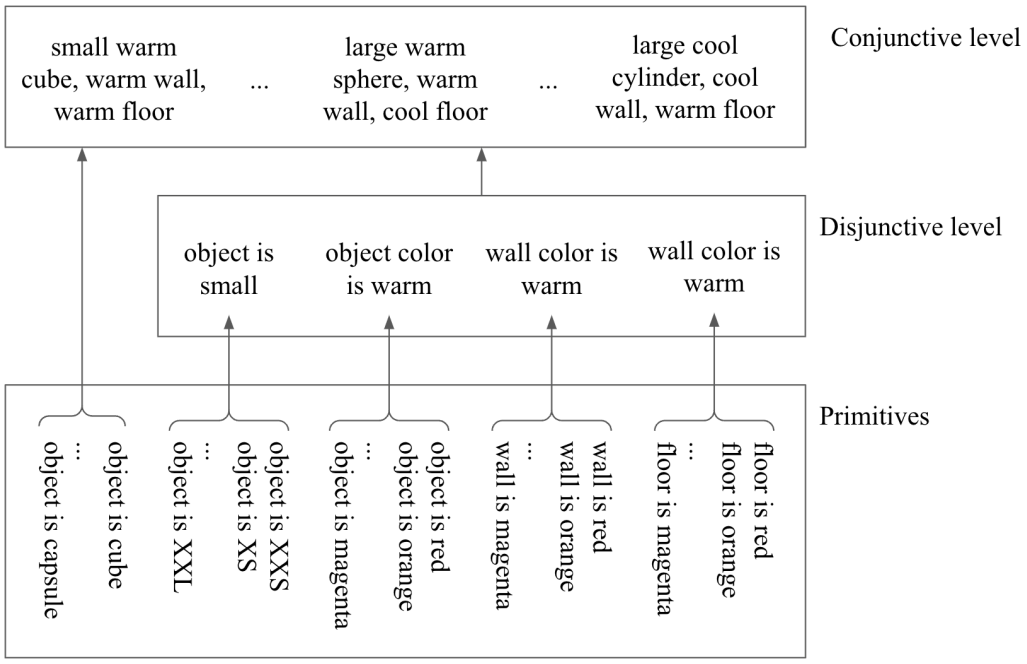

Figure 12: Ontology of training labels for the 3D Shapes dataset [Burgess and Kim, 2018].

The last inequality is because $y(1-y) < 1$, i.e., $\log\left(y(1-y)\right) < 0$.

In §2.3.1, we set a constraint on the kernel k such that $k(a,a) = \log(\frac{1}{\epsilon}+1)$. By Eq. 7, the entropy of $f(a)$ can be bounded as follows.

$$\mathbb{H}[f(a)] = \frac{1}{2}\log(2\pi e k_D(a,a)) \leq \frac{1}{2}\log(4\pi e \log(\frac{1}{\epsilon}+1)).$$

Hence there exists a lower bound on episteme for GPP. However, $\mathbb{H}[g(a)]$ can approach $-\infty$ because (1) variable y can be infinitely close to 0 or 1, and (2) $k_D(a,a)$ can also be infinitely close to 0, which means episteme has no finite upper bound.

In natural language, our analyses of episteme show that ignorance has a limit, but knowledge has no limit. This is a widely recognized idea, and it is also reflected in the words of Zhuangzi, a Chinese philosopher from the 4th century BCE: "Your life has a limit, but knowledge has none."

C Experiment details

In this section, we include details on experiment setups and additional results.

C.1 3D Shapes ontology

The ontology of training labels for the 3D Shapes dataset [Burgess and Kim, 2018] is illustrated in Figure 12. Images are generated from 6 ground truth independent primitives: 10 floor colors, 10 wall colors, 10 object colors, 8 scales, 4 shapes and 15 orientations of the shapes (orientation is excluded from the ontology since it's only distinguishable for cubes). The disjunctive level of the ontology groups together ranges of color and shape primitives into binary concepts: warm/cool and small/large, respectively. Concepts in the conjunctive level are the Cartesian product of concepts in the disjunctive level and the shape primitives.

C.2 Real-world OOD detection

In-distribution (ID) queries are sampled disjointly from the validation split of the ImageNet dataset[Russakovsky et al., 2015], where the probe observes 10 sets of Ds using 10 binary classification tasks defined by ImageNet superclasses. These superclasses are defined by building a tree using

the WordNet hierarchy [Miller, 1994] where the leaves of this tree are ImageNet classes (e.g., the superclass "dog" contains Chihuahua, Japanese Spaniel, Maltese, etc.). The ten classification tasks we use are: (1) dog vs snake, (2) fish vs lizard, (3) bird vs snake, (4) dog vs bird, (5) cat vs bird, (6) fish vs snake, (7) bird vs fish, (8) snake vs lizard, (9) cat vs dog,, (10) bird vs lizard.

Out-of-distribution (OOD) images are generated with pixel-wise uniform random noise. This noise is passed through the basis function ϕ to construct the OOD query.

C.3 Relations between judged probability, episteme and alea

In this section, we present more empirical results. The experiment setting is the same as §3.3.

First, we evaluate how judged probability and alea change as episteme increases, and how judged probability changes as alea increases. Figure 13 and Figure 14 show the results for GPP and LPE respectively. Each row corresponds to a different ground truth probability, which means the probability that an originally positive stimulus remains to have positive labels in observations, when we manually inject fuzziness to concepts in the 3D Shapes dataset [Burgess and Kim, 2018]. So in the ideal case, judged probability predictions should converge to the ground truth probability for stimuli that are originally positive. Each scattered point corresponds to the predictions on a stimulus that is originally positive.

GPP consistently produces rational uncertainty measures. There are no extreme judged probability predictions with low episteme. Alea converges to low values for 1.0 ground truth probability and higher values when the ground truth probability is 0.25 or 0.75, and alea converges to the highest values when the ground truth probability is 0.5. These are all expected since with 0.5 ground truth probability, the level fuzziness reaches the highest.

On the contrary, LPE tends to have more extreme predictions on judged probability no matter what the ground truth probability is. However, the average judged probability of LPE does get affected by the ground truth probability. For example, when the ground truth probability is 0.25, more masses of judged probability accumulate between 0 to 0.2. This means the average judged probability can be close to 0.25. Similarly, when the ground truth probability is 0.5, about half of the predictions of judged probability are between 0.8 to 1.0, and the other half are between 0.0 to 0.2. While this ensures the judged probability is close to ground truth probability if we average over all stimuli, the individual predictions cannot be used to evaluate the fuzziness of concepts.

Figure 15 shows AUROC, AUPRC and accuracy of GPP and LPE for different numbers of observations. These metrics are averaged over both positive and negative queries. Interestingly, even when the ground truth probability is 0.25 (only 1/4 of the positive examples remain positive), GPP can still achieve almost 1.0 AUROC and AUPRC. As expected, the accuracy of GPP is about 0.5 for 0.25 ground truth probability (since the judged probability predictions are mostly lower than 0.5 for the positive examples, and the negative examples are almost all correct). But because LPE does not always have low judged probability predictions even if ground truth probability is 0.25, its accuracy is higher than 0.5.

These results confirm the rationality and good performance of GPP as a valid probing method.

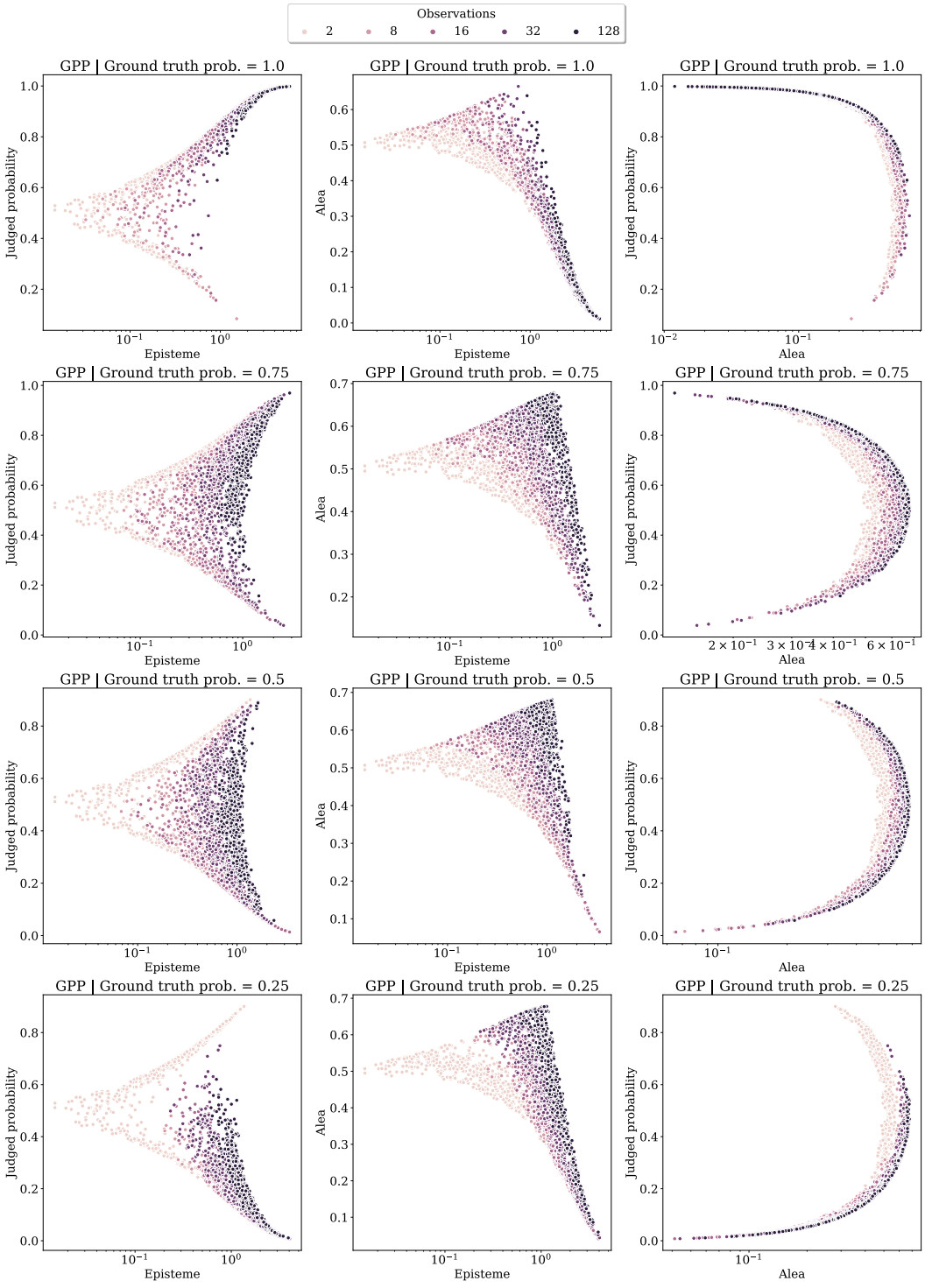

Figure 13: Relationships between judged probability, episteme and alea using GPP.

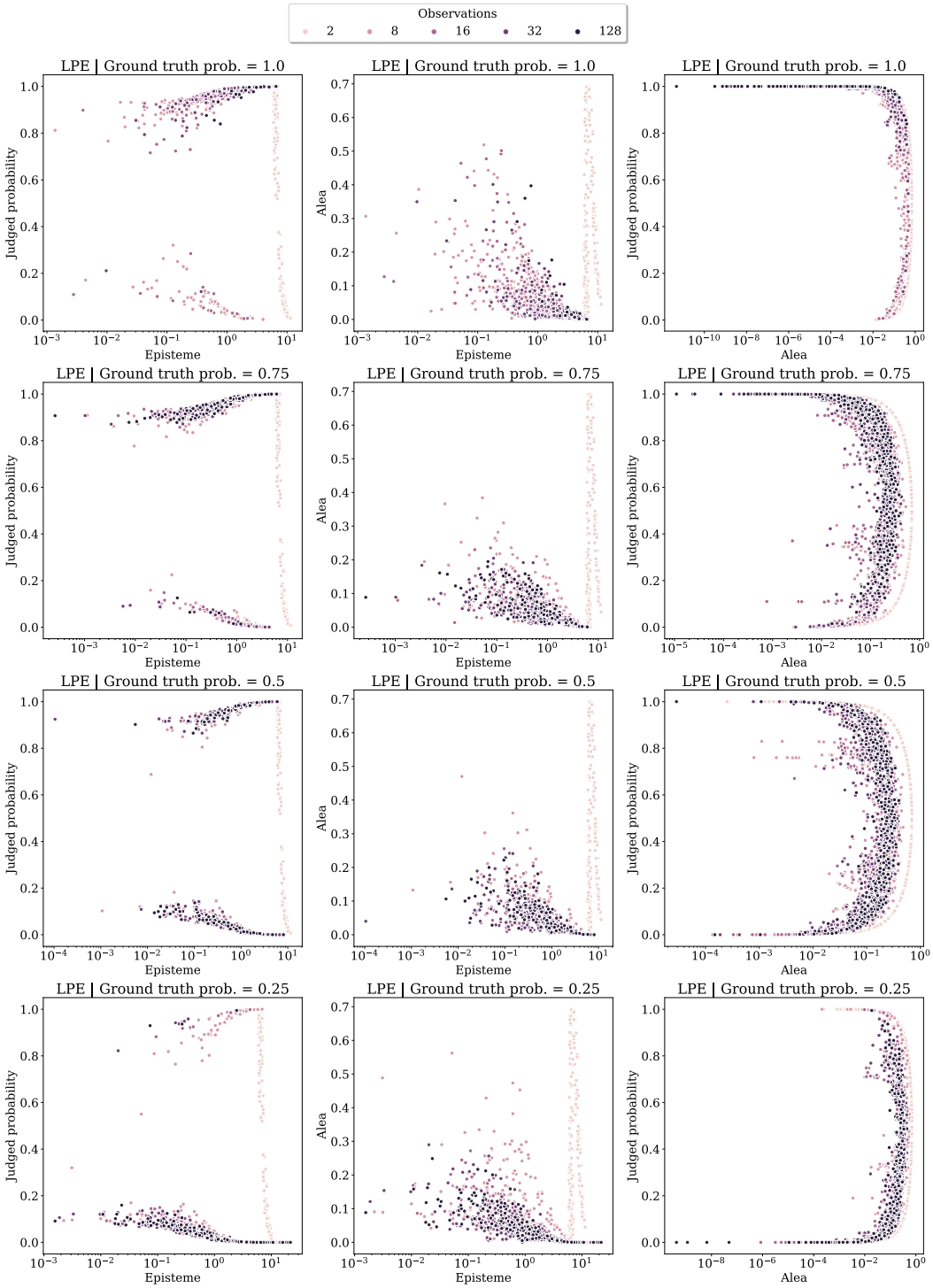

Figure 14: Relationships between judged probability, episteme and alea using LPE.

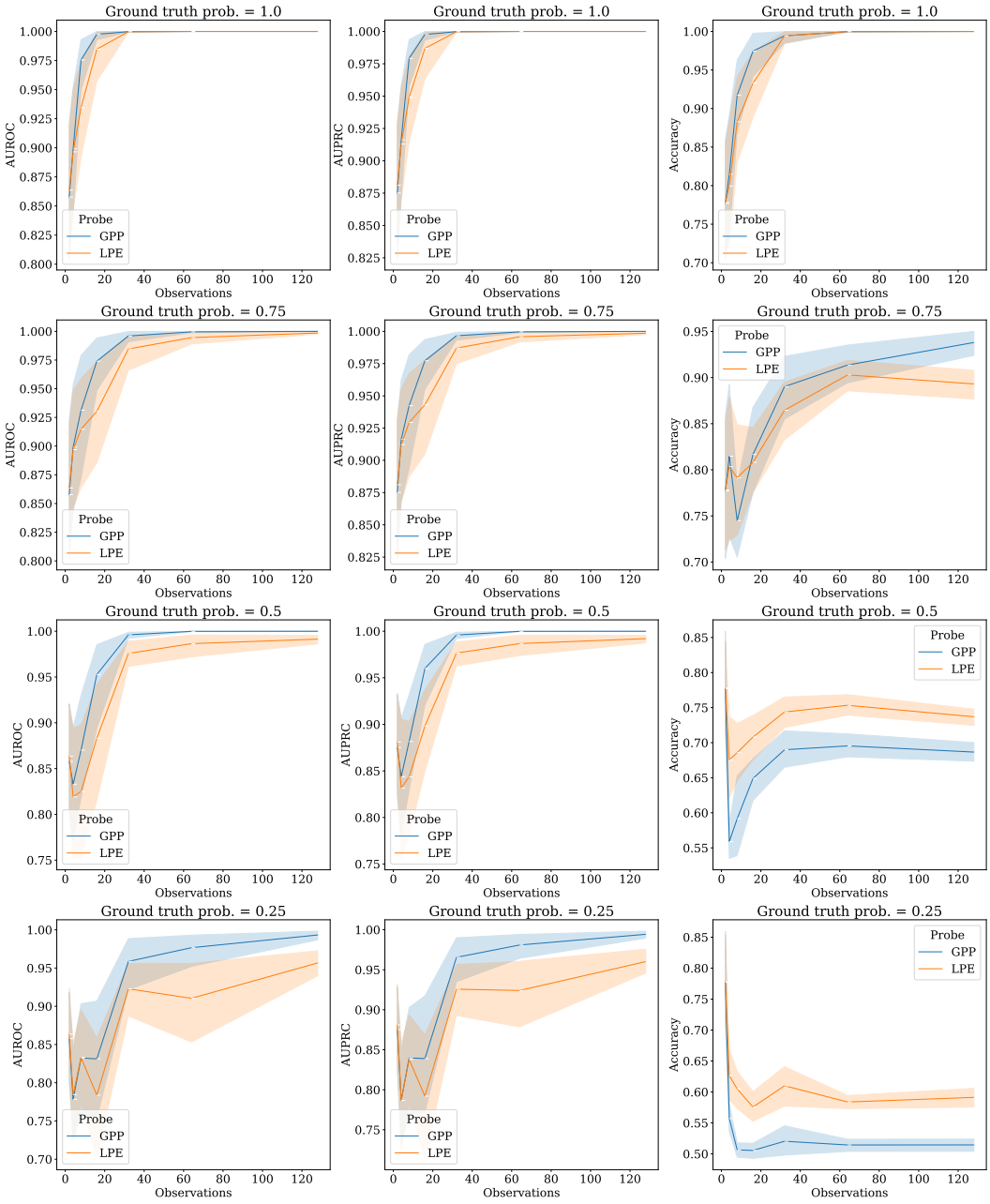

Figure 15: AUROC, AUPRC and accuracy of GPP and LPE using different numbers of observations.