Environment-Robust Representation Learning with Empirical Bayes

Yuli Slavutsky
Department of Statistics
Columbia University
New York, NY 10027, USA
ys3938@columbia.edu

Matthew Shen
Department of Statistics
Columbia University
New York, NY 10027, USA
ms7079@columbia.edu

Bohan Wu
Department of Statistics
Columbia University
New York, NY 10027, USA
bw2766@columbia.edu

David M. Blei
Departments of Statistics and Computer Science
Columbia University
New York, NY 10027, USA
david.blei@columbia.edu

Abstract

We consider multi-environment prediction problems. We assume the environments change the distribution of a latent variable, while the mechanisms generating observed covariates and targets remain stable conditional on that variable. For example, hospitals or clinical cohorts may differ in the prevalence of latent patient states, even though the relationships between those states, physiological measurements, and outcomes remain unchanged. Given a dataset from multiple environments, we formulate a Bayesian model for such problems and derive the corresponding variational objective. We show that this objective decomposes into per-environment terms and an additional cross-environment balancing term induced by the model's structure. We use an empirical Bayes method to set the prior and incorporate it into the objective. Based on this objective, we develop an amortized variational algorithm for posterior approximation, and use the resulting learned latent variables to form predictions in new environments. We study our approach through simulations and real-world studies of astronomical source identification, microbiome-based disease detection, and ICU sepsis prediction. Across these settings, our method outperforms previous approaches for prediction in new environments.

1 Introduction

Scientific prediction problems often involve data collected across heterogeneous environments. In astronomical source identification, for example, we observe measurements of objects across different regions of the sky, and the goal is to predict the object type. Different sky regions can have different prevalence of latent underlying source types [Humphrey et al., 2022], while the measurement process and the relationship between source types and labels are stable across regions. In microbiome studies, we observe microbial abundance profiles from patients across different body habitats or study cohorts, and the goal is to predict disease. Different environments can induce different mixture proportions over latent microbial states [Consortium, 2012], while sequencing and labeling mechanisms are stable across cohorts and habitats.

Motivated by these examples, we develop a probabilistic model for multi-environment prediction [Peters et al., 2016], where each environment captures variation in the distribution of latent variables. Formally, we assume that environments are sampled from $p(e)$ and for each environment, latent variables are sampled from $p(z \mid e)$. An outcome of interest y depends on z through $p(y \mid z)$. We

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

observe a potentially high-dimensional proxy variable x, sampled according to $p(x \mid z)$, where the mechanisms $p(x \mid z)$ and $p(y \mid z)$ remain stable across environments. This corresponds to the probabilistic graphical model in Figure 1.

We observe a training dataset $\mathcal{D} = \{\{(x_i^e, y_i^e)\}_{i=1}^{n_e}\}_{e \in \mathcal{E}_{\text{tr}}}$, drawn across a finite set of training environments $\mathcal{E}_{\text{tr}} = \{e_1, \ldots, e_m\}$. The probability of this dataset is

$$p(\mathcal{D}) = \prod_{j=1}^{m} p(e_j) \prod_{i=1}^{n_{e_j}} \int p(z_{ij} \mid e_j)\, p(y_{ij} \mid z_{ij})\, p(x_{ij} \mid z_{ij})\, dz_{ij}. \tag{1}$$

Our goal is to predict an outcome y for a new observation x, drawn according to $p(x \mid e = e')$, where e' is a new environment that was not observed during training. Since $e' \notin \mathcal{E}_{\text{tr}}$, the environment-specific Bayes target $p(y \mid x, e') = \int p(y \mid z)\, p(z \mid x, e')\, dz$ cannot in general be recovered. For this reason, we target the marginal Bayes rule $p(y \mid x)$, where

$$p(y \mid x) = \iint p(z, e \mid x)\, p(y \mid x, z, e)\, dz\, de = \iint p(z \mid x, e)\, p(e \mid x)\, p(y \mid z)\, dz\, de. \tag{2}$$

Our target, namely estimating $p(y \mid x)$ for an observation x from an unobserved environment, requires estimating three terms. The first term is

$$p(z \mid x, e) = \int p(y \mid x, e)\, p(z \mid x, y, e)\, dy. \tag{3}$$

In the astronomical source classification example, it corresponds to the distribution of latent source types z given object measurements x, and sky region e. Estimating this term requires averaging over the possible labels for the same observation x. In the data, however, each observed x is typically paired with only one realization of y. Consequently, the latent information directly inferred from an observed pair is label-conditioned: it corresponds to $p(z \mid x, y, e)$, rather than to the desired marginal quantity $p(z \mid x, e)$.

Note, however, that $p(y \mid x, e)$ does not include the latent z, which allows directly estimating it from the data. We approximate $p(z \mid x, y, e)$ and obtain $p(z \mid x, e)$ by marginalizing over $p(y \mid x, e)$. This is an empirical-Bayes [Efron, 2012] step: the prior distribution $p(z \mid x, e)$, which depends on the known covariates x but not y, is estimated using observed data.

The second term is $p(e \mid x)$. In our astronomical source classification example, it measures how plausible each observed sky region e is for measurements x. A naive estimate is the training indicator of whether x was observed in environment e, but this indicator is unavailable for observations from new environments. Such estimates are not observation-specific: they do not adapt to a new input x, and cannot estimate how likely each training environment is for that observation. To address this, we learn a predictor of $p(e \mid x)$ that estimates the probability of each training environment given the observation x.

The third term is the likelihood of each label y given a latent state z, that is $p(y \mid z)$. To this end, we introduce an amortized variational neural model in which z is inferred from the observed data, while $p(y \mid z)$ is parameterized directly as the conditional label distribution given the latent state.

Together, these ideas form the Empirical-Bayes Environment-Robust (EmBER) method. In what follows, we derive a variational objective and the resulting estimation algorithm.

We then show that EmBER can be interpreted as an *environment-balancing* method, namely a method whose objective combines environment-wise terms with a cross-environment regularization penalty. Unlike standard environment-balancing methods, this penalty is not pre-specified, but is induced by the probabilistic model.

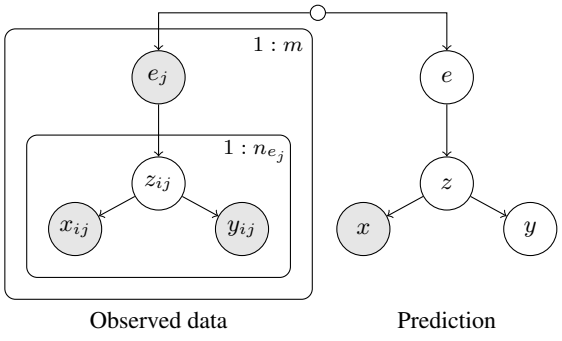


Figure 1: Probabilistic graphical model of the multi-environment data generative process. Observed variables shown in gray. The top dot encodes shared distribution across environments.

Finally, we evaluate EmBER on synthetic and real-world datasets, and show that it often improves predictive performance relative to other environment-balancing methods.

1.1 Related Work

A central line of work in out-of-distribution generalization seeks representations for which the predictive relationship with the target remains stable across environments. These methods typically optimize an objective of the form

$$\min_\theta \ \sum_{e\in\mathcal{E}_{\text{tr}}} \ell_e(f_\theta) + \lambda\,\mathcal{R}(f_\theta, \mathcal{E}_{\text{tr}}), \tag{4}$$

where $\ell_e(f_\theta)$ is the loss of the representation f_θ in environment e, and $\mathcal{R}$ encourages some notion of similar behavior across environments. This perspective underlies invariant representation learning methods such as invariant risk minimization and its variants [Arjovsky et al., 2019, Ahuja et al., 2020, Lu et al., 2021, Lin et al., 2022], causality-oriented robust methods [Shen et al., 2023], as well as invariant approaches to transfer learning [Rojas-Carulla et al., 2018]. Closely related methods, including risk extrapolation and calibration-based objectives, also fit this framework by favoring predictors whose risks or calibration levels remain stable across environments [Krueger et al., 2021, Wald et al., 2021, Shi et al., 2021].

In §4 we show that our approach belongs to this category of methods. However, rather than predefining a notion of stability across environments and enforcing it through a chosen penalty, we derive the balancing term from the assumed latent-variable model.

A complementary direction is to model environment-specific variation explicitly, rather than penalizing it away. Under different probabilistic assumptions, Slavutsky and Blei [2026] show that explicitly modeling the effect of the environment can improve robust prediction.

Other related lines of work target different notions of stability. These include distributionally robust optimization [Ben-Tal et al., 2013, Duchi et al., 2021, Duchi and Namkoong, 2021, Wei et al., 2023] and its group variants [Sagawa et al., 2019, Piratla et al., 2021], which optimize performance under worst-case perturbations, rather than the average performance over new environments.

Another related line is latent-variable representation learning across multiple conditions. Several methods based on variational autoencoders [Kingma and Welling, 2014] separate shared and condition-specific factors [Sohn et al., 2015, Klys et al., 2018, Lopez et al., 2018, Joy et al., 2020]. More recently, Wu et al. [2026] used empirical Bayes for multi-domain causal representation learning under a linear mixing model. However, these approaches impose independence, causal, or prior assumptions that do not match our assumed generative process.

In Appendix A we discuss broader connections beyond the most directly related literature, including causal representation learning beyond environment-balancing criteria, connections to mixed-effects models, and worst-case robustness approaches.

2 Multi-Environment Variational Learning

Our goal is to estimate the marginal predictive distribution in Equation (2).

For the true probability p, under the probabilistic graphical model in Figure 1 we have that

$$\log p(y \mid x) = \mathbb{E}_{p(z\mid x,y)}[\log p(y \mid z)] - \mathrm{KL}(p(z \mid x, y) \,\|\, p(z \mid x)), \tag{5}$$

where

$$p(z \mid x, y) = \int p(z \mid x, y, e)\, p(e \mid x, y)\, de \tag{6}$$

$$p(z \mid x) = \int p(z \mid x, e)\, p(e \mid x)\, de. \tag{7}$$

Since by Bayes' rule $p(e \mid x, y) \propto p(y \mid x, e)\, p(e \mid x)$, it suffices to estimate (i) the posterior $p(z \mid x, y, e)$, (ii) the conditional distribution of the label given the latent variable $p(y \mid z)$, (iii) the conditional distribution of the label given the observation and environment $p(y \mid x, e)$, and (iv) the conditional distribution of the environment $p(e \mid x)$.

Below, we formulate a single objective to estimate all these terms.

Let $q(z; x, y, e)$ be an environment-specific variational approximation to $p(z \mid x, y, e)$. Using the same marginalization identities as in Equations (6)–(7), this variational posterior induces the global posterior $q(z; x, y)$ and the label-marginal prior $q(z; x)$.

Given these induced distributions, the corresponding conditional distribution of y given x is

$$p_q(y \mid x) \coloneqq \int p(y \mid z)\, q(z; x)\, dz. \tag{8}$$

Our goal is to maximize the conditional log-likelihood

$$\frac{1}{|\mathcal{D}|} \sum_{e \in \mathcal{E}_{\text{tr}}} \sum_{i=1}^{n_e} \log p_q(y_i^e \mid x_i^e), \tag{9}$$

where, for each observation,

$$p_q(y_i^e \mid x_i^e) \coloneqq \int p(y_i^e \mid z)\, q(z; x_i^e)\, dz. \tag{10}$$

The conditional log-likelihood is generally intractable. However, for any variational family such that $q(z; x_i^e, y_i^e)$ is absolutely continuous with respect to $q(z; x_i^e)$, by Jensen's inequality[1] we have that

$$\begin{aligned} \log p_q(y_i^e \mid x_i^e) &= \log \int p(y_i^e \mid z)\, q(z; x_i^e)\, dz = \log \mathbb{E}_{q(z; x_i^e, y_i^e)} \left[\frac{p(y_i^e \mid z)\, q(z; x_i^e)}{q(z; x_i^e, y_i^e)} \right] \\ &\geq \mathbb{E}_{q(z; x_i^e, y_i^e)} \left[\log p(y_i^e \mid z) \right] - \mathrm{KL}\left(q(z; x_i^e, y_i^e) \,\|\, q(z; x_i^e) \right). \end{aligned} \tag{11}$$

Consequently, each term in the empirical conditional log-likelihood admits the lower bound in Equation (11).

We now specify the parameterized estimators that define the distributions appearing in this bound.

Parameterized Estimators Rather than optimizing separate variational parameters for each observation, we model the environment-specific variational posterior $q(z; x, y, e)$ using an amortized neural network g_θ, parameterized by θ. This network maps (x, y, e) to the mean and diagonal covariance entries of a Gaussian distribution, $g_\theta(x, y, e) = (\mu_\theta(x, y, e), \sigma_\theta(x, y, e))$.

That is, we define

$$q(z; x, y, e) = \mathcal{N}(\mu_\theta(x, y, e), \mathrm{diag}(\sigma_\theta(x, y, e))). \tag{12}$$

Next, we model the conditional distribution $p(y \mid z)$ with a neural predictor $f_\phi^{(1)}$, shared across environments. This defines the estimate $\hat{p}_\phi(y \mid z)$.

For estimation of the environment conditionals, we specify an auxiliary non-latent neural network $f_\varphi^{(0)}$ that takes (x, e) as input and estimates $\hat{p}_\varphi(y \mid x, e)$. Similarly, we define a network h_ψ that takes x as input and outputs an estimate $\hat{p}_\psi(e \mid x)$. Using Bayes' rule, these two observed-variable conditionals define

$$\hat{p}_{\varphi,\psi}(e \mid x, y) \propto \hat{p}_\varphi(y \mid x, e)\, \hat{p}_\psi(e \mid x). \tag{13}$$

Applying the marginalization identities in Equations (6) - (7) to the parameterized estimators yields

$$\hat{q}_{\theta,\varphi,\psi}(z; x, y) \coloneqq \int \hat{q}_\theta(z; x, y, e)\, \hat{p}_{\varphi,\psi}(e \mid x, y)\, de, \quad \hat{q}_{\theta,\varphi,\psi}(z; x) \coloneqq \int \hat{q}_{\theta,\varphi}(z; x, e)\, \hat{p}_\psi(e \mid x)\, de, \tag{14}$$

where

$$\hat{q}_{\theta,\varphi}(z; x, e) \coloneqq \int q_\theta(z; x, y, e)\, \hat{p}_\varphi(y \mid x, e)\, dy. \tag{15}$$

Optimization Objective Substituting these estimates into Equation (11) gives our objective

$$\widehat{\mathcal{L}}(\theta, \phi, \varphi, \psi) \coloneqq \frac{1}{|\mathcal{D}|} \sum_{e \in \mathcal{E}_{\text{tr}}} \sum_{i=1}^{n_e} \left[\mathbb{E}_{\hat{q}_{\theta,\varphi,\psi}(z; x_i^e, y_i^e)} \left[\log p_\phi(y_i^e \mid z) \right] - \mathrm{KL}\left(\hat{q}_{\theta,\varphi,\psi}(z; x_i^e, y_i^e) \,\|\, \hat{q}_{\theta,\varphi,\psi}(z; x_i^e) \right) \right]. \tag{16}$$

We learn all the parameters of these estimators jointly by optimizing Equation (16).

[1] This is a slight variation of the standard ELBO derivation.

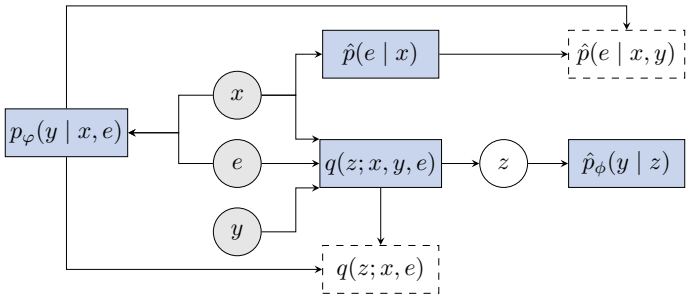


Figure 2: Schematic illustration of the training computations. Observed variables are gray circles. Blue rectangles denote quantities estimated directly by neural networks. Dashed rectangles denote derived quantities, computed from other estimates.

Note that $\hat{p}_\psi(e \mid x)$ is learned implicitly: it enters the objective only through the aggregation weights in $\hat{q}(z; x)$. In low-signal settings, adding a supervised environment-prediction term $\lambda_{\text{env}} {}^1\!/\!_{|\mathcal{D}|} \sum_{e \in \mathcal{E}_{\text{tr}}} \sum_{i=1}^{n_e} \log \hat{p}_\psi(e \mid x_i^e)$ may produce better estimates as it encourages the classifier to maximize the likelihood of the environments explicitly. We use this term only in the deliberately low-signal simulation in Section 5.1, where we set $\lambda_{\text{env}} = 1$; in all other experiments, we found no meaningful difference and set $\lambda_{\text{env}} = 0$.

Unlike reconstruction-based approaches, our objective does not require z to retain all the information in x, but rather the information relevant for predicting the scalar outcome y. Accordingly, in all experiments we set the dimension of z to one.

In practice, we approximate the expectation terms in Equation (16) with a single-sample Monte Carlo estimator using standard Gaussian reparametrization. The resulting amortized training procedure is summarized in Algorithm 1 and is illustrated in Figure 2.

Algorithm 1

Require: Dataset $\mathcal{D}$, initial values $(\theta_0, \phi_0, \varphi_0, \psi_0)$, number of iterations T, batch size b, step size η

for $t = 1, \ldots, T$ **do**
  Sample a minibatch $B = \{(x_i, y_i, e_i)\}_{i=1}^{b}$ from $\mathcal{D}$
  for each $(x_i, y_i, e_i) \in B$ **do**
    Compute $q(z; x_i, y_i, e_i) = g_{\theta_{t-1}}(x_i, y_i, e_i)$
    Sample $z_i \sim q(z; x_i, y_i, e_i)$ by Gaussian reparametrization (12)
    Evaluate the likelihood term $p_{\phi_{t-1}}(y_i \mid z_i)$ using $f^{(1)}_{\phi_{t-1}}(z_i)$
    For all $e \in \mathcal{E}_{\text{tr}}$ compute

$$p_{\varphi_{t-1}}(y_i \mid x_i, e) \text{ using } f^{(0)}_{\varphi_{t-1}}(x_i, e)$$
$$q(z; x_i, e) = \int q(z; x_i, y, e)\, p_{\varphi_{t-1}}(y \mid x_i, e)\, dy$$
$$\hat{p}_{\psi_{t-1}}(e \mid x_i) = h_{\psi_{t-1}}(x_i)$$
$$\hat{p}(e \mid x_i, y_i) \propto \hat{p}_{\psi_{t-1}}(e \mid x_i)\, p_{\varphi_{t-1}}(y_i \mid x_i, e)$$

    Compute $\hat{q}(z; x_i, y_i) = \sum_{e \in \mathcal{E}_{\text{tr}}} \hat{p}(e \mid x_i, y_i)\, q(z; x_i, y_i, e)$
    Compute $\hat{q}(z; x_i) = \sum_{e \in \mathcal{E}_{\text{tr}}} \hat{p}_{\psi_{t-1}}(e \mid x_i)\, q(z; x_i, e)$
  end for
  Form $\hat{\mathcal{L}}_B$ as the batch approximation of Equation (16)
  Update $(\theta_t, \phi_t, \varphi_t, \psi_t) = (\theta_{t-1}, \phi_{t-1}, \varphi_{t-1}, \psi_{t-1}) + \eta \nabla_{\theta,\phi,\varphi,\psi} \hat{\mathcal{L}}_B$
end for
return $(\theta_T, \phi_T, \varphi_T, \psi_T)$

2.1 Prediction

At prediction time, we observe a new datapoint x, but not its environment. Once training is complete with Algorithm 1, prediction for a datapoint x, possibly from a new environment e', proceeds by marginalizing over the training environments. Since e' is not observed at test time, we first compute,

for each $e \in \mathcal{E}_{\text{tr}}$, the environment-specific conditional prior $q(z; x, e) = \int q(z; x, y, e)\, p_\varphi(y \mid x, e)\, dy$ and the environment weights $\hat{p}_\psi(e \mid x) = h_\psi(x)$. We then form the EB-prior

$$\hat{q}(z; x) = \textstyle\sum_{e \in \mathcal{E}_{\text{tr}}} \hat{p}_\psi(e \mid x)\, q(z; x, e). \tag{17}$$

The predictive distribution for y is obtained by averaging over this aggregated distribution,

$$\hat{p}(y \mid x) = \textstyle\int \hat{p}_\phi(y \mid z)\, \hat{q}(z; x)\, dz = \sum_{e \in \mathcal{E}_{\text{tr}}} \hat{p}_\psi(e \mid x) \int \hat{p}_\phi(y \mid z)\, q(z; x, e)\, dz. \tag{18}$$

3 Extrapolation to New Environments

A central question is why the EmBER model can extrapolate to observation x from a new environment $e' \notin \mathcal{E}_{\text{tr}}$? The answer is that the learned distribution $\hat{p}_\psi(e \mid x)$ defines a prediction rule that adaptively averages over the per-environment effects: at test time, the model evaluates the latent representation induced by each training environment, and combines these contributions according to how plausible each of them is for the observed covariate x.

Specifically, Equation (18) can be interpreted as expressing the new observation in terms of its similarity to the training environments: it is a weighted average of environment-specific predictions across the training environments, with weights given by $\hat{p}_\psi(e \mid x)$.

This averaging of environment-specific predictions over observed environments is the empirical-Bayes step: the prior $\hat{q}(z \mid x)$ is not prespecified, but estimated from the data. For connections to EB g-modeling, see Laird [1978], especially Equation (3.3) on self-consistent priors.

To gain intuition for the resulting extrapolation, we next present a simple example illustrating the importance of explicitly modeling $p(e \mid x)$ when environments differ in the distribution of latent states.

Motivating Example: Explicit Modeling of Environments Let $e \sim p(e)$ be a random environment, and let $\pi_e \in [0, 1]$ be its environment-specific parameter. Assume that $z \mid e \sim \text{Bernoulli}(\pi_e)$ and

$$p(x \mid z = 1) = 1 + \alpha(2x - 1), \qquad p(x \mid z = 0) = 1 - \alpha(2x - 1), \tag{19}$$

where $0 < \alpha < 1$ is a known parameter. The label is a noisy copy of z:

$$p(y = 1 \mid z = 1) = 1 - \rho_y, \qquad p(y = 1 \mid z = 0) = \rho_y, \qquad 0 < \rho_y < 1/2. \tag{20}$$

As in the assumed model, z depends on e while $p(x \mid z)$ and $p(y \mid z)$ are stable across environments.

Suppose we observe m training environments $e_1, \ldots, e_m$, with environment-specific parameters $\pi_{e_1}, \ldots, \pi_{e_m}$. Even in an idealized setting where the environment-specific posteriors $p(z|x, e_j)$ are known, accurate prediction requires weighting training environments according to the observation x.

In this example, these weights are not uniform: large values of x are evidence for $z = 1$, which makes environments with larger π_{e_j} more plausible. Consequently, the correct conditional environment weights $p(e \mid x)$ yield convergence to the population predictive distribution at rate $m^{-1/2}$. In contrast, uniform weights, which ignore the dependence on x, converge to a biased limit. Formal statements and proofs of these results are provided in Propositions 2 and 3 in Appendix B. Figure 3 illustrates this phenomenon numerically. Additional details are provided in Appendix E.1.

This example illustrates that accurate prediction requires approximation of both the environment-specific latent distributions $p(z \mid x, e)$ and the observation-specific environment-weights $p(e \mid x)$, beyond naive empirical frequencies that yield non-adaptive uniform weights.

4 EmBER as an Environment-Balancing Method

We now show that EmBER can be framed as an environment-balancing method. Specifically, the expected value of the lower bound in Equation (11) decomposes into an average of environment-specific terms, and an additional cross-environment discrepancy term that compares the environment-specific variational components with their global counterparts. The next theorem makes this precise.

Theorem 1. *Let*

$$\mathcal{L}_e \coloneqq \mathbb{E}_{p(x,y|e)} \left[\mathbb{E}_{q(z;x,y,e)} [\log p(y \mid z)] - \text{KL}(q(z; x, y, e) \,\|\, q(z; x, e)) \right],$$

and denote $\mathcal{L} \coloneqq \mathbb{E}_{p(x,y)}[\mathcal{L}(x, y)]$. *Then,* $\mathcal{L} = \mathbb{E}_{p(e)}[\mathcal{L}_e] + R(q)$ *where*

$$R(q) \coloneqq \mathbb{E}_{p(x,y)} \Big[\mathbb{E}_{p(e|x,y)} [\text{KL}(q(z; x, y, e) \,\|\, q(z; x, y))] \Big] - \mathbb{E}_{p(x)} \Big[\mathbb{E}_{p(e|x)} [\text{KL}(q(z; x, e) \,\|\, q(z; x))] \Big].$$

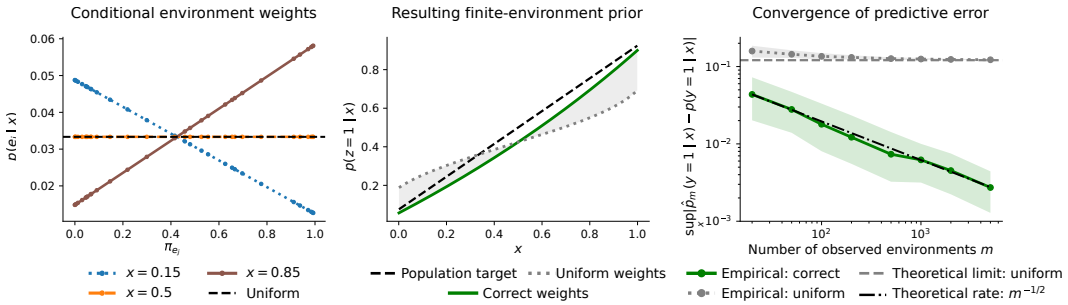


Figure 3: Illustration of the role of conditional environment weights. Left: weights $p(e_j \mid x)$ over observed training environments vary with x; larger values of x favor environments with larger π_{e_j}. Middle: these weights determine the finite-environment approximation to $p(z = 1 \mid x)$. Correct conditional weighting tracks the population target, whereas uniform weighting is biased. Right: as the number of observed environments increases, the predictor based on correct weights converges at $m^{-1/2}$ rate, while the uniformly weighted predictor converges to a biased limit.

The proof is provided in Appendix C.1. Theorem 1 shows that EmBER induces an environment-balancing term that follows directly from the latent-variable model. This balancing term $R(q)$ is composed from two discrepancy terms. The first compares the environment-specific posteriors $q(z; x, y, e)$ with the global posterior $q(z; x, y)$. The second compares the corresponding environment-specific priors $q(z; x, e)$ with the global prior $q(z; x)$. The balancing term is large when posterior discrepancy across environments exceeds the discrepancy already present in the corresponding priors.

The decomposition in Theorem 1 also shows why naive estimation of $\hat{p}(e \mid x, y)$ and $\hat{p}(e \mid x)$ directly from empirical frequencies in the training data is inadequate. In continuous settings, each observed pair (x_i, y_i) typically appears in only one environment e_i. Consequently, the empirical conditionals collapse to point masses yielding $\hat{q}(z; x_i, y_i) = q(z; x_i, y_i, e_i)$ and $\hat{q}(z; x_i) = q(z; x_i, e_i)$. As a result, both cross-environment discrepancy terms in $R(q)$ vanish, and the objective reduces to an average of environment-specific objectives. In this collapsed regime, the aggregation becomes trivial and the environment weights $p(e \mid x)$ and $p(e \mid x, y)$ are not learned or used during training.

However, this collapsed objective is incompatible with the intended test-time use. At test time, the environment label is unobserved, so prediction must aggregate the environment-specific quantities $q(z; x, e)$ using the observation-specific weights $\hat{p}(e \mid x)$. If training uses empirical point-mass weights, each observation is assigned only to its observed environment, so the model is never forced to learn how environment-specific components should be combined for a new x. Preserving the global objective therefore requires modeling the environment weights explicitly: $\hat{p}(e \mid x, y)$ determines how environment-specific posteriors are combined during training, while $\hat{p}(e \mid x)$ determines how environment-specific priors are combined during both training and prediction.

5 Empirical Results

We evaluate EmBER on synthetic and real-world multi-environment classification tasks. The synthetic experiments are designed to match the assumed structure. The real-world experiments test the same principle in settings with naturally occurring environments. We evaluate predictive performance using negative log-likelihood (NLL) and classification accuracy.

We compare EmBER with empirical risk minimization (ERM) and with environment-balancing methods: IRM [Arjovsky et al., 2019], VREx [Krueger et al., 2021], and Fishr [Rame et al., 2022] (see details in §1.1). Within each experiment, all methods use the same train-test environment split and model. We test each method's ability to form accurate predictions on held-out environments.

Across all 5 experiments, EmBER is the strongest method overall. In the parametric simulation, it achieves the lowest NLL and highest accuracy, with accuracy nearly matching the Bayes-optimal predictor, and remains best across the sensitivity and ablation studies. The same pattern holds in the colored MNIST simulation. On the real-world tasks, EmBER again performs best, with substantial improvements over all competitors in quasar-star and over all competitors except VREx

in microbiome. In the sepsis task, EmBER achieves the highest AUPRC and substantial AUROC gains over all methods except IRM. An additional method-agreement analysis shows that EmBER's mistakes are not concentrated on examples that are consistently easy for competing methods.

For each experiment, we perform one-sided t-tests comparing EmBER with the competing methods and apply the Benjamini–Yekutieli correction for multiple comparisons. Additional details for all experiments are provided in Appendix E, and additional results in Appendix D. Code is available at `https://github.com/mattyshen/EmBER`.

5.1 Parametric Simulation

We first study a controlled setting. For each environment j, we assign a branch index b_j and draw an environment-specific branch center $m_j = \Delta b_j + \phi_0 + \delta_j$ with $\delta_j \sim \mathcal{N}(0, \sigma_\delta^2)$. Then, for observation i in environment j, we sample $z_{ij} \mid e_j \sim \mathcal{N}(m_j, \sigma_z^2)$. The observed covariates are generated as

$$x_{ij,1} = \cos(z_{ij}) + \varepsilon_{ij,1}, \qquad x_{ij,2} = \sin(z_{ij}) + \varepsilon_{ij,2}, \qquad x_{ij,3} = \rho z_{ij} + \varepsilon_{ij,3}, \tag{21}$$

and $x_{ij,k} = \varepsilon_{ij,k}$ for $k = 4, \ldots, d$. The first two coordinates reveal the phase of z_{ij} but not its branch, while the third coordinate provides a noisy branch cue. The target is generated by the stable latent mechanism $y_{ij} \mid z_{ij} \sim \text{Bernoulli}\left(\sigma\left(\beta \sin(z_{ij}/2)\right)\right)$.

Method	NLL ↓	Accuracy ↑	PV (NLL)
Bayes-x	0.233 ± 0.004	0.934 ± 0.002	–
EmBER	$\mathbf{0.319 \pm 0.069}$	$\mathbf{0.928 \pm 0.003}$	–
ERM	0.652 ± 0.005	0.577 ± 0.009	0.015
IRM	0.654 ± 0.006	0.573 ± 0.012	0.015
V-REx	0.640 ± 0.026	0.572 ± 0.016	0.015
Fishr	0.652 ± 0.006	0.582 ± 0.011	0.015

Table 1: Parametric simulation results. EmBER performs best among all learned methods and nearly matches optimal-Bayes accuracy.

Sensitivity and ablation analysis We vary branch-cue strength ρ, the number of training environments m_{train}, number of observations per environment n, and data dimension d. We compare ERM, IRM, VREx, Fishr, EmBER, and EmBER variants that replace learned components with oracle values.

Results Table 1 shows results for the baseline parameters. As an oracle reference, we report the Bayes-optimal predictor (Bayes-x). EmBER yields the highest accuracy and the lowest NLL among all non-oracle methods. The improvement is statistically significant relative to all methods with $p = 0.015$. Across 8 sensitivity studies and 6 ablation studies, EmBER remains best overall among methods that we implemented. Detailed results and analysis of these studies are reported in Appendix D.1.

5.2 Colored MNIST Simulation

To illustrate the setting targeted by our method, we construct a variant of the colored MNIST experiment [Arjovsky et al., 2019] in which the environment affects the data only through the distribution of a latent prototype variable $z \in \{1, 2, 3, 4\}$. We generate data according to

$$e \sim p(e), \qquad z \sim p(z|e), \qquad y \sim p(y|z), \qquad d \sim p(d|z), \qquad c \sim p(c|z), \qquad x \sim p(x|d, c),$$

where $y \in \{0, 1\}$ is the binary label, $d \in \{0, \ldots, 9\}$ is the digit, $c \in \{\text{red}, \text{green}\}$ is the color, and x is the corresponding colored image. The environment affects only the latent prototype mixture $p(z|e)$; it does not directly alter the label mechanism or image rendering. Given z, the digit identity is sampled from a prototype-specific distribution over digits, so d carries no environment-specific information beyond z.

The four prototypes have stable color and label tendencies: $z \in \{1, 2\}$ is mostly associated with label 0, while $z \in \{3, 4\}$ is mostly associated with label 1; $z \in \{1, 3\}$ is mostly red, while $z \in \{2, 4\}$ is mostly green. Each prototype also has a distinct, overlapping distribution over digit identities

Method	NLL ↓	Accuracy ↑	PV (NLL)
EmBER	$\mathbf{0.519 \pm 0.077}$	$\mathbf{0.787 \pm 0.071}$	–
ERM	0.863 ± 0.090	0.650 ± 0.021	0.004
IRM	0.572 ± 0.007	0.747 ± 0.017	0.203
V-REx	0.605 ± 0.019	0.684 ± 0.011	0.140
Fishr	0.917 ± 0.061	0.639 ± 0.020	0.004

Table 2: Colored MNIST simulation results. EmBER performs best among all methods.

$p(d|z)$. Thus, training environments differ only through the prototype mixture $p(z|e)$. We generate six training environments and one test environment whose prototype mixture reverses the induced color-label association. Additional details are provided in Appendix E.3.

Results Table 2 shows that EmBER yields the lowest NLL and highest accuracy among all methods.

5.3 Quasar-Star Classification

We use the publicly available Sloan Digital Sky Survey dataset [Ahumada et al., 2020]. The covariates x are photometric color features, the label y is quasar status, and the environment e is the sky-position group defined by binned absolute Galactic latitude $|b|$.

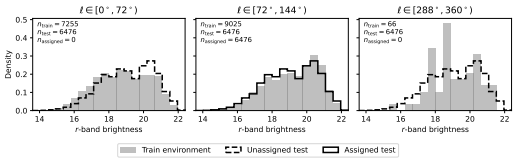


Figure 4: Quasar-star r-band brightness distributions by training environment. Gray histograms show the distribution in each training environment indexed by ℓ. For each test point, we assign the point to $\arg\max_e \hat{p}(e \mid x_{\text{test}})$. All test points are assigned to training environment $\ell \in [72°, 144°)$, whose distribution most closely aligns with the test distribution; unassigned environments show poorer agreement with the test distribution.

Results Table 3 shows that EmBER yields the highest accuracy and the lowest NLL among all methods. The improvement is statistically significant relative to all methods with $p < 0.001$.

Figure 5 examines the subset of test points on which EmBER disagrees with competing methods and reports how often the competing methods classify these points correctly. No competing method performs consistently well on this subset, indicating that EmBER's errors are not concentrated on examples that are systematically easy for the other methods.

We additionally analyze how the environments are assigned by the learned model. For all test points, EmBER assigns the largest estimated weight $\hat{p}(e \mid x)$ to the same training environment. Figure 4 compares the train and test distributions for the most variable feature; analogous comparisons for additional features are reported in Appendix Figure 7. The results show that EmBER assigns highest weight to the environment whose distribution is most similar to the test distribution.

Method	NLL ↓	Accuracy ↑	PV (NLL)
EmBER	0.221 ± 0.019	0.921 ± 0.007	–
ERM	0.390 ± 0.001	0.897 ± 0.001	< 0.001
Fishr	0.384 ± 0.004	0.894 ± 0.001	< 0.001
IRM	0.383 ± 0.009	0.891 ± 0.003	< 0.001
VREx	0.382 ± 0.009	0.892 ± 0.003	< 0.001

Table 3: Quasar-star classification results. EmBER performs best among all methods.

Method	NLL ↓	Accuracy ↑	PV (NLL)
EmBER	0.850 ± 0.112	0.591 ± 0.027	–
ERM	1.685 ± 0.136	0.587 ± 0.016	0.002
Fishr	1.449 ± 0.226	0.582 ± 0.021	0.008
IRM	1.548 ± 0.224	0.588 ± 0.011	0.007
VREx	0.903 ± 0.087	0.571 ± 0.014	0.543

Table 4: Microbiome classification results. EmBER performs best among all methods.

5.4 Microbiome Classification

We use the publicly available MicrobiomeHD datasets [Duvallet et al., 2017] for a colorectal-cancer microbiome classification task. Here, covariates x are genus-level abundance profiles, y is colorectal-cancer status, and environments e are study cohorts. Appendix E.5 provides additional details.

Results Table 4 shows that EmBER achieves the highest accuracy and the lowest NLL among all methods. The improvement is statistically significant with $p < 0.01$ relative to all methods except VREx. The results in Figure 6 further show that here as well, no competing method performs consistently well on the subset of test points on which EmBER is incorrect.

5.5 ICU Sepsis Prediction

We evaluate all methods on a multi-environment sepsis prediction task derived from the PhysioNet 2019 Challenge data [Reyna et al., 2020]. Each example corresponds to one ICU stay. Environments are defined by the public data source and the recorded ICU unit, serving as coarse hospital-systems. Covariates x are demographic variables and summaries of vital signs and laboratory measurements from the first 24 hours, labels y indicate whether a patient becomes septic later in the stay, and environments e are hospital-system groups, defined by the public data source and recorded ICU unit. For each ICU stay, we use measurements from the first 24 hours and predict whether the patient becomes septic later in the stay. Patients whose sepsis label becomes positive during the first 24 hours are excluded. The covariates consist of demographic variables and summaries of early vital signs and laboratory measurements.

Method	AUROC ↑	AUPRC ↑	PV (AUROC)
EmBER	**0.740 ± 0.008**	**0.179 ± 0.008**	–
ERM	0.692 ± 0.017	0.153 ± 0.013	0.003
Fishr	0.697 ± 0.014	0.156 ± 0.008	0.002
IRM	0.724 ± 0.012	0.169 ± 0.012	0.112
VREx	0.697 ± 0.012	0.154 ± 0.011	0.001

Table 5: Sepsis prediction results. EmBER performs best among all methods.

We train on five source-unit environments and evaluate on a held-out source-unit environment. Since sepsis is rare, we evaluate predictions using AUROC and AUPRC.

Results Table 5 shows that EmBER achieves the highest AUROC and AUPRC among all methods. The improvement is statistically significant with $p < 0.004$ relative to all methods except IRM.

6 Discussion

We studied multi-environment prediction problems where environments change the distributions of the latent variables, while the covariate and label mechanisms remain stable. Our goal was to recover the marginal Bayes rule $p(y \mid x)$ by averaging over latent and environment-specific variables.

Our main contribution is an empirical-Bayes variational method for carrying out this marginalization. Our analysis shows that the corresponding objective induces a specific balancing term: the relevant penalty is a discrepancy term that is implied by the latent-variable model itself. The analysis also shows that generalization to new environments requires explicit estimation of the environment effect through $\hat{p}(e \mid x)$ and $\hat{p}(e \mid x, y)$. The empirical results show that our method produces substantially better predictive performance compared to other methods.

Our model assumes that environments affect the latent distribution $p(z \mid e)$, while $p(x \mid z)$ and $p(y \mid z)$ remain stable. Since z is unobserved, these assumptions cannot be directly tested from observational data alone. However, their predictive implications can be assessed on held-out environments: strong predictive performance supports the adequacy of these assumptions for generalization.

Unlike other variational approaches, our method does not require prior specification, instead it uses the data to inform the prior. However, computing the aggregate distributions $\hat{q}(z \mid x, y)$ and $\hat{q}(z \mid x)$ increases the computational cost. A promising direction for future work is approximating these aggregates with a VampPrior-inspired mixture [Tomczak and Welling, 2018] built from a smaller number of learned pseudo-observations.

References

[1] Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *International Conference on Machine Learning*, pages 145–155. PMLR, 2020.

[2] Kartik Ahuja, Divyat Mahajan, Yixin Wang, and Yoshua Bengio. Interventional causal representation learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 372–407. PMLR, 2023. URL https://proceedings.mlr.press/v202/ahuja23a.html.

[3] Romina Ahumada, Carlos Allende Prieto, Andrés Almeida, Friedrich Anders, Scott F Anderson, Brett H Andrews, Borja Anguiano, Riccardo Arcodia, Eric Armengaud, Marie Aubert, et al. The 16th data release of the sloan digital sky surveys: first release from the apogee-2 southern survey and full release of eboss spectra. *The Astrophysical Journal Supplement Series*, 249(1): 3, 2020.

[4] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[5] Yuval Atzmon, Felix Kreuk, Uri Shalit, and Gal Chechik. A causal view of compositional zero-shot recognition. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/1010cedf85f6a7e24b087e63235dc12e-Abstract.html.

[6] Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

[7] Johann Brehmer, Pim de Haan, Phillip Lippe, and Taco S. Cohen. Weakly supervised causal representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 38319–38331, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/fa567e2b2c870f8f09a87b6e73370869-Abstract-Conference.html.

[8] The Human Microbiome Project Consortium. A framework for human microbiome research. *Nature*, 486(7402):215–221, 2012.

[9] Eugene Demidenko. *Mixed models: theory and applications with R*. John Wiley & Sons, 2013.

[10] John C Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021.

[11] John C Duchi, Peter W Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 46(3): 946–969, 2021.

[12] Claire Duvallet, Sean M Gibbons, Thomas Gurry, Rafael A Irizarry, and Eric J Alm. Meta-analysis of gut microbiome studies identifies disease-specific and shared responses. *Nature communications*, 8(1):1784, 2017.

[13] Bradley Efron. *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. Cambridge University Press, 2012. ISBN 9780511761362. doi: 10.1017/CBO9780511761362.

[14] Yorgos Felekis, Theodoros Damoulas, and Paris Giampouras. Distributionally robust causal abstractions. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 29930–29980. PMLR, 2026. URL https://proceedings.mlr.press/v306/felekis26a.html.

[15] Daniel Greenfeld and Uri Shalit. Robust learning with the Hilbert-Schmidt independence criterion. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3759–3768. PMLR, 2020. URL https://proceedings.mlr.press/v119/greenfeld20a.html.

[16] A Humphrey, W Kuberski, J Bialek, N Perrakis, W Cools, N Nuyttens, H Elakhrass, and PAC Cunha. Machine-learning classification of astronomical sources: estimating f1-score in the absence of ground truth. *Monthly Notices of the Royal Astronomical Society: Letters*, 517(1): L116–L120, 2022.

[17] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, pages 336–351. PMLR, 2022.

[18] Tom Joy, Sebastian M Schmon, Philip HS Torr, N Siddharth, and Tom Rainforth. Capturing label characteristics in vaes. *arXiv preprint arXiv:2006.10102*, 2020.

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2014.

[20] Jack Klys, Jake Snell, and Richard Zemel. Learning latent subspaces in variational autoencoders. *Advances in Neural Information Processing Systems*, 31, 2018.

[21] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021.

[22] Nan Laird. Nonparametric maximum likelihood estimation of a mixing distribution. *Journal of the American Statistical Association*, 73(364):805–811, 1978.

[23] Yong Lin, Hanze Dong, Hao Wang, and Tong Zhang. Bayesian invariant risk minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16021–16030, 2022.

[24] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021.

[25] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, 2018.

[26] Chaochao Lu, Yuhuai Wu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. Invariant causal representation learning for out-of-distribution generalization. In *International Conference on Learning Representations*, 2021.

[27] Chengzhi Mao, Augustine Cha, Amogh Gupta, Hao Wang, Junfeng Yang, and Carl Vondrick. Generative interventions for causal learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3947–3956, 2021. doi: 10.1109/CVPR46437.2021.00394. URL `https://openaccess.thecvf.com/content/CVPR2021/html/Mao_Generative_Interventions_for_Causal_Learning_CVPR_2021_paper.html`.

[28] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Number 37 in Monographs on Statistics and Applied Probability. Chapman and Hall, London, 2 edition, 1989.

[29] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(5):947–1012, 2016.

[30] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Focus on the common good: Group distributional robustness follows. In *International Conference on Learning Representations*, 2021.

[31] Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 18347–18377. PMLR, 2022.

[32] Matthew A Reyna, Christopher S Josef, Russell Jeter, Supreeth P Shashikumar, M Brandon Westover, Shamim Nemati, Gari D Clifford, and Ashish Sharma. Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019. *Critical care medicine*, 48 (2):210–217, 2020.

[33] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *Journal of Machine Learning Research*, 19(36):1–34, 2018.

[34] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

[35] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021. doi: 10.1109/JPROC.2021.3058954.

[36] Xinwei Shen, Peter Bühlmann, and Armeen Taeb. Causality-oriented robustness: exploiting general additive interventions. *arXiv preprint arXiv:2307.10299*, 2023.

[37] Yuge Shi, Jeffrey Seely, Philip HS Torr, Narayanaswamy Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

[38] Giora Simchoni and Saharon Rosset. Using random effects to account for high-cardinality categorical features and repeated measures in deep neural networks. *Advances in Neural Information Processing Systems*, 34:25111–25122, 2021.

[39] Giora Simchoni and Saharon Rosset. Integrating random effects in deep neural networks. *Journal of Machine Learning Research*, 24(156):1–57, 2023.

[40] Yuli Slavutsky and David Blei. Robust representation learning through explicit environment modeling. *https://arxiv.org/abs/2604.26128*, 2026.

[41] Yuli Slavutsky, Sebastian Salazar, and David M Blei. Neural generalized mixed-effects models. *arXiv preprint arXiv:2604.10976*, 2026.

[42] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in Neural Information Processing Systems*, 28, 2015.

[43] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International conference on artificial intelligence and statistics*, pages 1214–1223. PMLR, 2018.

[44] M-N Tran, Nghia Nguyen, David Nott, and Robert Kohn. Bayesian deep net glm and glmm. *Journal of Computational and Graphical Statistics*, 29(1):97–113, 2020.

[45] Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. On calibration and out-of-domain generalization. *Advances in neural information processing systems*, 34:2215–2227, 2021.

[46] Jiaheng Wei, Harikrishna Narasimhan, Ehsan Amid, Wen-Sheng Chu, Yang Liu, and Abhishek Kumar. Distributionally robust post-hoc classifiers under prior shifts. In *International Conference on Learning Representations*, 2023.

[47] Bohan Wu, Julius von Kügelgen, and David M Blei. Multi-domain empirical bayes for linearly-mixed causal representations. *arXiv e-prints*, pages arXiv–2603, 2026.

[48] Yunyang Xiong, Hyunwoo J Kim, and Vikas Singh. Mixed effects neural networks (menets) with applications to gaze estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7743–7752, 2019.

A Broader Related Work

In this section, we discuss broader connections beyond the most directly related literature in §1.1.

Causal representation learning. Beyond environment-balancing criteria-based methods, causal representation learning has been used to improve generalization across environments by identifying latent causal factors underlying high-dimensional observations [Schölkopf et al., 2021]. Causally motivated approaches have improved generalization under shifts involving previously unseen combinations of attributes [Atzmon et al., 2020], changes in experimental conditions [Greenfeld and Shalit, 2020], and shifts in context [Mao et al., 2021]. However, identifying causal representations often requires additional information or structural assumptions, such as paired observations before and after interventions [Brehmer et al., 2022] or interventions on latent variables [Ahuja et al., 2023].

Our approach similarly models observed variables through latent factors, but does not aim to identify causal variables, causal relationships, or intervention effects. Instead, we assume that environments change $p(z \mid e)$ while $p(x \mid z)$ and $p(y \mid z)$ remain stable. EmBER then learns to marginalize over environment-specific latent distributions to predict in new environments, without requiring identification of the latent variables as causal factors.

Mixed-effects models. Mixed-effects models capture group-specific variation through random effects while maintaining shared components [McCullagh and Nelder, 1989, Demidenko, 2013]. Estimating the parameters of the random-effects distribution by maximizing the marginal likelihood can be viewed as an empirical-Bayes procedure, in which the prior is learned from the data. Neural mixed-effects models extend this framework to nonlinear relationships [Simchoni and Rosset, 2021, 2023, Xiong et al., 2019, Tran et al., 2020, Slavutsky et al., 2026].

In EmBER, environments similarly index different distributions $p(z \mid e)$. The key distinction is that EmBER does not prescribe a parametric relationship between the observed covariates and the environment. Instead, it flexibly estimates $p(e \mid x)$ using a neural classifier and uses these observation-specific probabilities to aggregate the environment-specific latent distributions when predicting in new environments.

Worst-case robustness. Distributionally robust optimization (DRO) addresses distribution shifts by minimizing the worst-case expected loss over a set of plausible distributions [Ben-Tal et al., 2013, Duchi et al., 2021, Duchi and Namkoong, 2021, Wei et al., 2023]. Group DRO extends this approach to predefined groups, seeking robustness to changes in their relative prevalence [Sagawa et al., 2019, Piratla et al., 2021]. Related approaches address such shifts through reweighting high-loss observations [Liu et al., 2021] or balancing groups through subsampling [Idrissi et al., 2022]. More recently, Felekis et al. [2026] developed a DRO framework for learning causal abstractions that are robust to shifts in exogenous noise distributions.

Unlike these worst-case approaches, EmBER targets the marginal Bayes predictor under a probabilistic model of environment variation. Rather than optimizing against an adversarial distribution, it targets best possible prediction on average over environments sampled from the same distribution.

B Formal Analysis of the Motivating Example

Here we provide the formal statements for the motivating example in Section 3.

Recall the setting in Section 3: $e \sim p(e)$, where $\pi_e \in [0,1]$ are environment-specific parameters. We assume that $z \mid e \sim \text{Bernoulli}(\pi_e)$ and

$$p(x \mid z=1) = 1+\alpha(2x-1), \qquad p(x \mid z=0) = 1-\alpha(2x-1), \tag{22}$$

where $0 < \alpha < 1$ is a known parameter, and that the label is a noisy copy of z:

$$p(y=1 \mid z=1) = 1-\rho_y, \qquad p(y=1 \mid z=0) = \rho_y, \qquad 0 < \rho_y < 1/2. \tag{23}$$

Suppose we observe m training environments $e_1, \ldots, e_m$, with environment-specific parameters $\pi_{e_1}, \ldots, \pi_{e_m}$. Let $\mu \coloneqq \mathbb{E}_{p(e)}[\pi_e]$ and $\hat{\mu}_m \coloneqq \frac{1}{m}\sum_{j=1}^{m} \pi_{e_j}$. Then, the global prior and its estimator (obtained by averaging over the m observed environments) are

$$p(z=1|x) = \frac{\mu p(x|z=1)}{\mu p(x|z=1)+(1-\mu)p(x|z=0)}, \quad \hat{p}_m(z=1|x) = \frac{\hat{\mu}_m p(x|z=1)}{\hat{\mu}_m p(x|z=1)+(1-\hat{\mu}_m)p(x|z=0)}.$$

Since $p(y \mid z)$ is known, the predictive distribution and its finite-environment estimator are

$$p(y=1 \mid x) = \rho_y + (1-2\rho_y)p(z=1 \mid x), \quad \hat{p}_m(y=1 \mid x) = \rho_y + (1-2\rho_y)\hat{p}_m(z=1 \mid x). \tag{24}$$

Correct conditional environment weights. We first analyze the estimator $\hat{p}_m(y=1 \mid x)$ obtained by averaging over the observed environments using the correct conditional environment weights. In this case, the only source of error is the error from replacing the population average $\mu = \mathbb{E}_{p(e)}[\pi_e]$ by its empirical version $\hat{\mu}_m$. As a result, the finite-environment Bayes estimator $\hat{p}_m(y=1 \mid x)$ converges to the population predictive distribution $p(y=1 \mid x)$ at a rate of $m^{-1/2}$. This is formalized in the next proposition.

Proposition 2. *For every $\epsilon > 0$ and $\delta \in (0,1)$, to guarantee*

$$\Pr\left(\sup_{x\in[0,1]} |\hat{p}_m(y=1 \mid x) - p(y=1 \mid x)| \le \epsilon\right) \ge 1-\delta, \tag{25}$$

it is sufficient that

$$m \ge \frac{(1-2\rho_y)^2\kappa^2}{2\epsilon^2}\log\frac{2}{\delta}, \qquad \textit{with} \quad \kappa \coloneqq \frac{1+\alpha}{1-\alpha}. \tag{26}$$

The proof is provided in Appendix C.2.

The same conclusion, however, does not hold if we replace the conditional environment weights by uniform weights over the observed environments. We analyze this case next.

Uniform environment weights. Consider now the approximation obtained by replacing the conditional environment weights with uniform weights over the observed training environments. This changes the estimator from a plug-in estimate of $f_x(\mu)$ to an empirical average of $f_x(\pi_{e_j})$. Because f_x is nonlinear whenever $x \neq 1/2$, these two quantities have different population limits in general. The next proposition formalizes this bias.

Proposition 3. *Assume that $\mathrm{Var}_{p(e)}(\pi_e) > 0$. Then, for every fixed $x \in [0,1]$ such that $x \neq 1/2$,*

$$\tilde{p}_m(y=1|x) \longrightarrow \rho_y + (1-2\rho_y)\,\mathbb{E}_{p(e)}\left[\frac{\pi_e\, p(x|z=1)}{\pi_e\, p(x|z=1) + (1-\pi_e)\, p(x|z=0)}\right] \tag{27}$$

almost surely, and the limiting value is different from $p(y=1|x)$. Consequently, for every

$$0 < \epsilon < \left|\rho_y + (1-2\rho_y)\mathbb{E}_{p(e)}\left[\frac{\pi_e\, p(x|z=1)}{\pi_e\, p(x|z=1) + (1-\pi_e)\, p(x|z=0)}\right] - p(y=1|x)\right|, \tag{28}$$

we have that as $m \to \infty$,

$$\Pr\left(|\tilde{p}_m(y=1|x) - p(y=1|x)| > \epsilon\right) \longrightarrow 1. \tag{29}$$

The proof is provided in Appendix C.3.

C Proofs

C.1 Proof of Theorem 1

Proof. First, note that

$$\mathbb{E}_{q(z|x,y)}[\log p(y|z)] = \mathbb{E}_{p(e|x,y)}\mathbb{E}_{q(z|x,y,e)}\left[\log p(y|z)\right]. \tag{30}$$

Second, by definition

$$\mathrm{KL}(q(z|x,y,e) \,\|\, q(z|x)) = \mathbb{E}_{q(z|x,y,e)}\left[\log\frac{q(z|x,y,e)}{q(z|x)}\right]. \tag{31}$$

Taking expectation with respect to $p(e|x,y)$ and inserting $q(z|x,y)$, we get

$$\begin{aligned}\mathbb{E}_{p(e|x,y)}[\mathrm{KL}(q(z|x,y,e) \,\|\, q(z|x))] =& \mathbb{E}_{p(e|x,y)}\mathbb{E}_{q(z|x,y,e)}\left[\log\frac{q(z|x,y,e)}{q(z|x,y)}\right] \\ &+ \mathbb{E}_{q(z|x,y)}\left[\log\frac{q(z|x,y)}{q(z|x)}\right] \\ =& \mathbb{E}_{p(e|x,y)}[\mathrm{KL}(q(z|x,y,e) \,\|\, q(z|x,y))] + \mathrm{KL}(q(z|x,y) \,\|\, q(z|x)).\end{aligned} \tag{32}$$

Rearranging the last equation, we have

$$\begin{aligned}\mathrm{KL}(q(z|x,y)\,\|\,q(z|x)) =&\mathbb{E}_{p(e|x,y)}[\mathrm{KL}(q(z|x,y,e)\,\|\,q(z|x))] \\ &- \mathbb{E}_{p(e|x,y)}[\mathrm{KL}(q(z|x,y,e)\,\|\,q(z|x,y))].\end{aligned} \tag{33}$$

Substituting Equations (30) and (33) into $\mathcal{L}(x,y)$ we get

$$\begin{aligned}\mathcal{L}(x,y) =&\mathbb{E}_{p(e|x,y)}\Big[\mathbb{E}_{q(z|x,y,e)}[\log p(y|z)] \\ &- \mathrm{KL}(q(z|x,y,e)\,\|\,q(z|x))\Big] + \mathbb{E}_{p(e|x,y)}[\mathrm{KL}(q(z|x,y,e)\,\|\,q(z|x,y))].\end{aligned} \tag{34}$$

Averaging over $p(x,y)$ and using $p(x,y)p(e|x,y) = p(e)p(x,y|e)$ yields

$$\begin{aligned}\mathcal{L} =&\mathbb{E}_{p(x,y)}\Big[\mathbb{E}_{p(e|x,y)}\Big[\mathbb{E}_{q(z|x,y,e)}[\log p(y|z)] - \mathrm{KL}(q(z|x,y,e)\,\|\,q(z|x))\Big]\Big] \\ &+ \mathbb{E}_{p(x,y)}\Big[\mathbb{E}_{p(e|x,y)}[\mathrm{KL}(q(z|x,y,e)\,\|\,q(z|x,y))]\Big] &(35)\\ =&\mathbb{E}_{p(e)}\Big[\mathbb{E}_{p(x,y|e)}\Big[\mathbb{E}_{q(z|x,y,e)}[\log p(y|z)] &(36)\\ &- \mathrm{KL}(q(z|x,y,e)\,\|\,q(z|x))\Big]\Big] + \mathbb{E}_{p(x,y)}\Big[\mathbb{E}_{p(e|x,y)}[\mathrm{KL}(q(z|x,y,e)\,\|\,q(z|x,y))]\Big]. &(37)\end{aligned}$$

Now, for each fixed environment e, by adding and subtracting $q(z|x,e)$, we get

$$\begin{aligned}\mathbb{E}_{p(x,y|e)}[\mathrm{KL}(q(z|x,y,e)\,\|\,q(z|x))] =&\mathbb{E}_{p(x,y|e)}\mathbb{E}_{q(z|x,y,e)}\left[\log\frac{q(z|x,y,e)}{q(z|x)}\right] &(38)\\ =&\mathbb{E}_{p(x,y|e)}\mathbb{E}_{q(z|x,y,e)}\left[\log\frac{q(z|x,y,e)}{q(z|x,e)}\right] &(39)\\ &+ \mathbb{E}_{p(x,y|e)}\mathbb{E}_{q(z|x,y,e)}\left[\log\frac{q(z|x,e)}{q(z|x)}\right].\end{aligned}$$

Note that the first term is

$$\mathbb{E}_{p(x,y|e)}[\mathrm{KL}(q(z|x,y,e)\,\|\,q(z|x,e))]. \tag{40}$$

For the second term, since

$$q(z|x,e) = \mathbb{E}_{p(y|x,e)}[q(z|x,y,e)], \tag{41}$$

we have

$$\begin{aligned}&\mathbb{E}_{p(x,y|e)}\mathbb{E}_{q(z|x,y,e)}\left[\log\frac{q(z|x,e)}{q(z|x)}\right] &(42)\\ &= \mathbb{E}_{p(x|e)}\mathbb{E}_{p(y|x,e)}\mathbb{E}_{q(z|x,y,e)}\left[\log\frac{q(z|x,e)}{q(z|x)}\right] &(43)\\ &= \mathbb{E}_{p(x|e)}\mathbb{E}_{q(z|x,e)}\left[\log\frac{q(z|x,e)}{q(z|x)}\right] &(44)\\ &= \mathbb{E}_{p(x|e)}[\mathrm{KL}(q(z|x,e)\,\|\,q(z|x))]. &(45)\end{aligned}$$

Substituting Equations (40) and (45) back into Equation (38) yields

$$\begin{aligned}\mathbb{E}_{p(x,y|e)}[\mathrm{KL}(q(z|x,y,e)\,\|\,q(z|x))] =&\mathbb{E}_{p(x,y|e)}[\mathrm{KL}(q(z|x,y,e)\,\|\,q(z|x,e))] \\ &+ \mathbb{E}_{p(x|e)}[\mathrm{KL}(q(z|x,e)\,\|\,q(z|x))],\end{aligned} \tag{46}$$

and substituting into Equation (37),

$$\begin{aligned}\mathcal{L} = \mathbb{E}_{p(e)}&\left[\mathbb{E}_{p(x,y|e)}\Big[\mathbb{E}_{q(z|x,y,e)}[\log p(y|z)] - \mathrm{KL}(q(z|x,y,e)\,\|\,q(z|x,e))\Big] - \mathbb{E}_{p(x|e)}[\mathrm{KL}(q(z|x,e)\,\|\,q(z|x))]\right] \\ &+ \mathbb{E}_{p(x,y)}\Big[\mathbb{E}_{p(e|x,y)}[\mathrm{KL}(q(z|x,y,e)\,\|\,q(z|x,y))]\Big].\end{aligned} \tag{47}$$

Therefore, we have

$$\begin{aligned}\mathcal{L} =&\mathbb{E}_{p(e)}\Big[\mathcal{L}_e\Big] + \mathbb{E}_{p(x,y)}\Big[\mathbb{E}_{p(e|x,y)}[\mathrm{KL}(q(z|x,y,e) \,\|\, q(z|x,y))]\Big] \\ &- \mathbb{E}_{p(e)}\Big[\mathbb{E}_{p(x|e)}[\mathrm{KL}(q(z|x,e) \,\|\, q(z|x))]\Big].\end{aligned} \tag{48}$$

Using $p(x)p(e|x) = p(e)p(x|e)$ in the last term gives

$$\begin{aligned}\mathcal{L} =&\mathbb{E}_{p(e)}\Big[\mathcal{L}_e\Big] + \mathbb{E}_{p(x,y)}\Big[\mathbb{E}_{p(e|x,y)}[\mathrm{KL}(q(z|x,y,e) \,\|\, q(z|x,y))]\Big] \\ &- \mathbb{E}_{p(x)}\Big[\mathbb{E}_{p(e|x)}[\mathrm{KL}(q(z|x,e) \,\|\, q(z|x))]\Big].\end{aligned} \tag{49}$$

□

C.2 Proof of Proposition 2

Proof. For fixed x, denote

$$f_x(a) \coloneqq \frac{ap(x \mid z=1)}{ap(x \mid z=1) + (1-a)p(x \mid z=0)}. \tag{50}$$

Then,

$$p(z=1 \mid x) = f_x(\mu), \qquad \hat{p}_m(z=1 \mid x) = f_x(\hat{\mu}_m). \tag{51}$$

We have that

$$f'_x(a) = \frac{p(x \mid z=1)p(x \mid z=0)}{\left(ap(x \mid z=1) + (1-a)p(x \mid z=0)\right)^2}. \tag{52}$$

Since

$$1-\alpha \le p(x \mid z=u) \le 1+\alpha, \qquad u \in \{0,1\}, \tag{53}$$

we have

$$\sup_{x\in[0,1]} \sup_{a\in[0,1]} |f'_x(a)| \le \frac{1+\alpha}{1-\alpha} = \kappa. \tag{54}$$

Therefore,

$$\sup_{x\in[0,1]} |\hat{p}_m(z=1 \mid x) - p(z=1 \mid x)| \le \kappa|\hat{\mu}_m - \mu|. \tag{55}$$

Since

$$\hat{p}_m(y=1 \mid x) - p(y=1 \mid x) = (1-2\rho_y)\left(\hat{p}_m(z=1 \mid x) - p(z=1 \mid x)\right), \tag{56}$$

it follows that

$$\sup_{x\in[0,1]} |\hat{p}_m(y=1 \mid x) - p(y=1 \mid x)| \le (1-2\rho_y)\kappa|\hat{\mu}_m - \mu|. \tag{57}$$

Finally, since $\pi_{e_1}, \ldots, \pi_{e_m} \in [0,1]$ are iid, Hoeffding's inequality gives

$$\Pr\left(|\hat{\mu}_m - \mu| > t\right) \le 2\exp(-2mt^2). \tag{58}$$

Substituting

$$t = \frac{\epsilon}{(1-2\rho_y)\kappa} \tag{59}$$

completes the proof. □

C.3 Proof of Proposition 3

Proof. As in the previous proposition, for fixed x, denote

$$f_x(a) \coloneqq \frac{ap(x \mid z=1)}{ap(x \mid z=1) + (1-a)p(x \mid z=0)}. \tag{60}$$

Then,

$$p(z=1 \mid x) = f_x(\mu), \qquad \tilde{p}_m(z=1 \mid x) = \frac{1}{m}\sum_{j=1}^{m} f_x(\pi_{e_j}). \tag{61}$$

By the law of large numbers,

$$\tilde{p}_m(z = 1 \mid x) \longrightarrow \mathbb{E}_{p(e)}[f_x(\pi_e)] \tag{62}$$

almost surely. Moreover,

$$f_x''(a) = -\frac{2p(x \mid z = 1)p(x \mid z = 0)\left(p(x \mid z = 1) - p(x \mid z = 0)\right)}{\left(ap(x \mid z = 1) + (1 - a)p(x \mid z = 0)\right)^3}. \tag{63}$$

Since

$$p(x \mid z = 1) - p(x \mid z = 0) = 2\alpha(2x - 1), \tag{64}$$

we have $f_x''(a) \neq 0$ for all $a \in [0, 1]$ whenever $x \neq 1/2$. Therefore, f_x is strictly convex or strictly concave. Since $\mathrm{Var}_{p(e)}(\pi_e) > 0$, Jensen's inequality is strict:

$$\mathbb{E}_{p(e)}[f_x(\pi_e)] \neq f_x(\mathbb{E}_{p(e)}[\pi_e]) = f_x(\mu). \tag{65}$$

Therefore, as $m \to \infty$,

$$\tilde{p}_m(z = 1 \mid x) \longrightarrow \mathbb{E}_{p(e)}[f_x(\pi_e)] \neq p(z = 1 \mid x). \tag{66}$$

Since

$$\tilde{p}_m(y = 1 \mid x) = \rho_y + (1 - 2\rho_y)\tilde{p}_m(z = 1 \mid x), \tag{67}$$

and

$$p(y = 1 \mid x) = \rho_y + (1 - 2\rho_y)p(z = 1 \mid x), \tag{68}$$

with $1 - 2\rho_y > 0$, it follows that

$$\tilde{p}_m(y = 1 \mid x) \longrightarrow \rho_y + (1 - 2\rho_y)\mathbb{E}_{p(e)}[f_x(\pi_e)] \neq p(y = 1 \mid x). \tag{69}$$

□

D Additional Empirical Results

D.1 Sensitivity and Ablation Studies

In this section we evaluate the robustness of EmBER through sensitivity and ablation studies in the parametric simulation. Starting from the baseline setting in Section 5.1, we vary one data-generating parameter at a time: the strength of the observed branch cue ρ, the number of training environments m, the number of observations per environment n, and the observed covariate dimension d. We compare EmBER with ERM, IRM, V-REx, and Fishr, and include the optimal-Bayes oracle reference that uses the optimal predictor based on the observed covariates.

We also evaluate EmBER variants that replace selected learned components with oracle or uniform alternatives, in order to isolate the role of the learned environment weights and the learned environment-specific posteriors. These additional variants are detailed in Table 6.

Variant	Name	$p(e\|x)$ train	$p(e\|x)$ test	Posterior train	Posterior test
1	EmBER (learned-weights-learned-posteriors)	learned	learned	learned	learned
2	EmBER (learned-train-weights-uniform-test-weights)	learned	uniform	learned	learned
3	EmBER (learned-train-weights-oracle-test-weights)	learned	oracle	learned	learned
4	EmBER (oracle-train-weights-oracle-test-weights)	oracle	oracle	learned	learned
5	EmBER (learned-weights-oracle-posteriors)	learned	learned	oracle	oracle
6	EmBER (uniform-train-weights-uniform-test-weights)	uniform	uniform	learned	learned
7	EmBER (uniform-weights-oracle-posteriors)	uniform	uniform	oracle	oracle

Table 6: EmBER variants used in the ablation and sensitivity studies.

Sensitivity analysis results Tables 7-14 show that across the sensitivity studies, EmBER remains the best method among the implementable methods in NLL, and usually also in accuracy.

Increasing the *branch-cue strength* leads to improvement in all methods, especially V-REx. This is true also for EmBER. The NLL of EmBER improves from the baseline value, and its accuracy reaches the optimal-Bayes accuracy.

Increasing the *number of training environments* brings EmBER closest to the optimal-Bayes: for $m = 12$ and $m = 36$, EmBER has NLL 0.254 and 0.252, compared with the optimal-Bayes 0.232 and 0.234. At $m = 36$ the baselines also improve substantially, so the relative advantage of EmBER becomes much smaller.

Increasing the *number of examples per environment* also helps EmBER: at $n = 1000$, EmBER nearly matches the optimal-Bayes in accuracy and NLL, while maintaining a large gap over the competing methods.

Varying the *data dimension* shows that at $d = 50$ and $d = 100$, EmBER is still clearly best among the implementable methods, but the gap to the optimal-Bayes widens. At $d = 1000$, EmBER still has lower NLL than ERM, IRM, V-REx, and Fishr, but the accuracy of all methods collapses.

Runtime is affected most strongly by the number of training environments: EmBER increases from about 36 seconds at $m = 12$ to 173 seconds at $m = 36$, whereas changes in data dimension have much smaller effects on the relative runtime.

Ablation analysis results Variant 3, which uses learned training weights but oracle test weights, is almost identical to the optimal-Bayes predictor in accuracy and often improves slightly over EmBER in NLL. The slight improvement indicates that even when the approximation of $p(e \mid x)$ learned by EmBER is not perfect, the approximation of the per-environment posteriors is relatively accurate. Variant 4, which uses oracle weights at both train and test time, is usually even closer to optimal-Bayes, indicating that the remaining gap between EmBER and Bayes-x is largely due to imperfect estimation of the environment weights.

By contrast, variants 2 and 6 consistently yield inferior performance: replacing the test weights by uniform weights, or using uniform weights during both training and testing, brings performance close to the ERM/IRM/V-REx/Fishr range. The same holds for variant 7, showing that even when oracle posteriors are used, as long as uniform weights are used, the performance is poor. Variant 5 shows similar trends in the low- and moderate-dimensional settings, despite using oracle posteriors, which suggests that the way environment-specific components are aggregated is central.

Together, the ablations support the main modeling claim: EmBER's gains come from learning observation-specific environment weights and using them consistently for aggregation.

D.1.1 Sensitivity to Branch-Cue Strength

Method	NLL ↓	Accuracy ↑	ECE ↓	Time (s)	PV (NLL)
Bayes-x	0.231 ± 0.004	0.935 ± 0.001	0.002 ± 0.000	3.293 ± 0.414	–
EmBER	**0.301 ± 0.071**	**0.935 ± 0.002**	0.099 ± 0.082	17.563 ± 0.694	–
ERM	0.620 ± 0.012	0.572 ± 0.016	0.119 ± 0.028	4.304 ± 0.773	0.024
IRM	0.606 ± 0.028	0.593 ± 0.032	0.115 ± 0.026	6.068 ± 0.966	0.024
V-REx	0.580 ± 0.055	0.644 ± 0.097	0.122 ± 0.037	5.173 ± 0.676	0.024
Fishr	0.603 ± 0.012	0.602 ± 0.031	**0.089 ± 0.033**	6.030 ± 0.626	0.024
EmBER–variant 2	0.656 ± 0.015	0.614 ± 0.036	0.152 ± 0.092	17.526 ± 0.670	–
EmBER–variant 3	0.292 ± 0.071	0.935 ± 0.001	0.087 ± 0.087	0.000 ± 0.000	–
EmBER–variant 4	0.242 ± 0.004	0.935 ± 0.001	0.010 ± 0.004	136.658 ± 8.059	–
EmBER–variant 5	0.753 ± 0.083	0.633 ± 0.011	0.325 ± 0.067	207.051 ± 1.671	–
EmBER–variant 6	0.664 ± 0.009	0.571 ± 0.022	0.064 ± 0.039	19.698 ± 1.036	–
EmBER–variant 7	0.674 ± 0.009	0.589 ± 0.048	0.092 ± 0.031	178.919 ± 1.296	–

Table 7: Sensitivity and ablation results for branch-cue strength: $\rho = 0.15$.

Method	NLL ↓	Accuracy ↑	ECE ↓	Time (s)	PV (NLL)
Bayes-x	0.231 ± 0.004	0.935 ± 0.001	0.002 ± 0.000	3.335 ± 0.267	–
EmBER	$\mathbf{0.297 \pm 0.067}$	$\mathbf{0.935 \pm 0.002}$	0.095 ± 0.080	17.390 ± 0.678	–
ERM	0.576 ± 0.026	0.626 ± 0.042	0.108 ± 0.056	4.602 ± 0.333	0.026
IRM	0.551 ± 0.036	0.652 ± 0.048	0.123 ± 0.018	5.946 ± 0.956	0.026
V-REx	0.544 ± 0.058	0.692 ± 0.081	$\mathbf{0.084 \pm 0.026}$	5.117 ± 0.405	0.026
Fishr	0.561 ± 0.032	0.658 ± 0.026	0.093 ± 0.032	6.458 ± 0.686	0.026
EmBER–variant 2	0.657 ± 0.021	0.578 ± 0.066	0.142 ± 0.086	17.677 ± 0.736	–
EmBER–variant 3	0.292 ± 0.066	0.935 ± 0.001	0.088 ± 0.081	0.000 ± 0.000	–
EmBER–variant 4	0.242 ± 0.003	0.935 ± 0.001	0.011 ± 0.004	133.081 ± 9.282	–
EmBER–variant 5	0.753 ± 0.082	0.628 ± 0.021	0.326 ± 0.065	196.114 ± 4.056	–
EmBER–variant 6	0.650 ± 0.014	0.573 ± 0.011	0.106 ± 0.058	19.060 ± 0.292	–
EmBER–variant 7	0.666 ± 0.010	0.553 ± 0.066	0.146 ± 0.034	170.055 ± 0.875	–

Table 8: Sensitivity and ablation results for branch-cue strength: $\rho = 0.20$.

D.1.2 Sensitivity to Number of Training Environments

Method	NLL ↓	Accuracy ↑	ECE ↓	Time (s)	PV (NLL)
Bayes-x	0.232 ± 0.005	0.935 ± 0.002	0.003 ± 0.002	3.306 ± 0.703	–
EmBER	$\mathbf{0.254 \pm 0.003}$	$\mathbf{0.932 \pm 0.002}$	$\mathbf{0.026 \pm 0.005}$	36.465 ± 1.990	–
ERM	0.572 ± 0.015	0.629 ± 0.003	0.099 ± 0.031	6.148 ± 1.002	0.003
IRM	0.544 ± 0.026	0.663 ± 0.034	0.113 ± 0.032	11.869 ± 1.147	0.006
V-REx	0.549 ± 0.051	0.675 ± 0.047	0.119 ± 0.021	7.431 ± 0.218	0.011
Fishr	0.527 ± 0.047	0.706 ± 0.070	0.105 ± 0.015	12.114 ± 0.616	0.011
EmBER–variant 2	0.639 ± 0.028	0.594 ± 0.074	0.104 ± 0.001	37.461 ± 2.338	–
EmBER–variant 3	0.247 ± 0.004	0.934 ± 0.002	0.016 ± 0.005	0.000 ± 0.000	–
EmBER–variant 4	0.242 ± 0.006	0.935 ± 0.002	0.009 ± 0.002	302.468 ± 5.353	–
EmBER–variant 5	0.674 ± 0.023	0.593 ± 0.081	0.186 ± 0.071	638.964 ± 19.802	–
EmBER–variant 6	0.674 ± 0.006	0.573 ± 0.015	0.039 ± 0.016	37.607 ± 1.260	–
EmBER–variant 7	0.694 ± 0.011	0.573 ± 0.064	0.110 ± 0.014	519.677 ± 4.796	–

Table 9: Sensitivity and ablation results for number of training environments: $m = 12$.

Method	NLL ↓	Accuracy ↑	ECE ↓	Time (s)	PV (NLL)
Bayes-x	0.234 ± 0.005	0.934 ± 0.002	0.003 ± 0.000	2.867 ± 0.036	–
EmBER	$\mathbf{0.252 \pm 0.005}$	$\mathbf{0.932 \pm 0.002}$	$\mathbf{0.021 \pm 0.003}$	173.110 ± 2.214	–
ERM	0.265 ± 0.009	0.931 ± 0.004	0.035 ± 0.006	10.531 ± 0.139	0.056
IRM	0.312 ± 0.007	0.926 ± 0.003	0.105 ± 0.008	53.156 ± 0.952	0.024
V-REx	0.296 ± 0.008	0.931 ± 0.002	0.094 ± 0.006	21.215 ± 0.475	0.024
Fishr	0.273 ± 0.002	0.930 ± 0.002	0.047 ± 0.007	39.614 ± 0.927	0.035
EmBER–variant 2	0.641 ± 0.016	0.643 ± 0.013	0.097 ± 0.006	172.446 ± 3.578	–
EmBER–variant 3	0.249 ± 0.006	0.933 ± 0.002	0.017 ± 0.005	0.000 ± 0.000	–
EmBER–variant 4	0.241 ± 0.005	0.934 ± 0.002	0.011 ± 0.002	1486.899 ± 45.133	–
EmBER–variant 5	0.661 ± 0.001	0.639 ± 0.003	0.187 ± 0.057	4185.989 ± 205.771	–
EmBER–variant 6	0.673 ± 0.006	0.573 ± 0.041	0.053 ± 0.027	160.284 ± 0.768	–
EmBER–variant 7	0.688 ± 0.000	0.615 ± 0.011	0.109 ± 0.011	3227.361 ± 30.513	–

Table 10: Sensitivity and ablation results for number of training environments: $m = 36$.

D.1.3 Sensitivity to Number of Training Examples per Environment

Method	NLL ↓	Accuracy ↑	ECE ↓	Time (s)	PV (NLL)
Bayes-x	0.233 ± 0.004	0.934 ± 0.002	0.002 ± 0.001	6.090 ± 0.769	–
EmBER	$\mathbf{0.250 \pm 0.006}$	$\mathbf{0.932 \pm 0.002}$	$\mathbf{0.019 \pm 0.005}$	35.630 ± 0.307	–
ERM	0.546 ± 0.003	0.665 ± 0.038	0.100 ± 0.038	8.766 ± 0.835	< 0.001
IRM	0.544 ± 0.038	0.658 ± 0.027	0.110 ± 0.028	12.393 ± 0.189	0.007
V-REx	0.509 ± 0.081	0.740 ± 0.098	0.095 ± 0.007	9.420 ± 0.916	0.032
Fishr	0.510 ± 0.031	0.713 ± 0.054	0.126 ± 0.036	12.524 ± 0.180	0.007
EmBER–variant 2	0.654 ± 0.005	0.546 ± 0.055	0.155 ± 0.003	34.958 ± 0.631	–
EmBER–variant 3	0.246 ± 0.005	0.933 ± 0.002	0.015 ± 0.004	0.000 ± 0.000	–
EmBER–variant 4	0.243 ± 0.005	0.934 ± 0.002	0.006 ± 0.004	285.632 ± 2.888	–
EmBER–variant 5	0.662 ± 0.000	0.639 ± 0.001	0.268 ± 0.030	465.074 ± 9.839	–
EmBER–variant 6	0.681 ± 0.002	0.571 ± 0.005	0.045 ± 0.011	39.347 ± 0.542	–
EmBER–variant 7	0.688 ± 0.000	0.589 ± 0.057	0.081 ± 0.056	402.842 ± 8.071	–

Table 11: Sensitivity and ablation results for training examples per environment: $n = 1000$.

D.1.4 Sensitivity to Data Dimension

Method	NLL ↓	Accuracy ↑	ECE ↓	Time (s)	PV (NLL)
Bayes-x	0.234 ± 0.002	0.934 ± 0.001	0.002 ± 0.001	3.438 ± 0.587	–
EmBER	$\mathbf{0.366 \pm 0.101}$	$\mathbf{0.907 \pm 0.007}$	0.126 ± 0.099	17.741 ± 0.333	–
ERM	0.685 ± 0.003	0.570 ± 0.002	0.071 ± 0.001	3.988 ± 0.340	0.035
IRM	0.688 ± 0.001	0.564 ± 0.003	0.057 ± 0.005	5.992 ± 0.755	0.035
V-REx	0.688 ± 0.004	0.561 ± 0.005	$\mathbf{0.052 \pm 0.006}$	5.470 ± 0.996	0.035
Fishr	0.689 ± 0.000	0.570 ± 0.001	0.077 ± 0.002	5.720 ± 0.314	0.035
EmBER–variant 2	0.666 ± 0.012	0.560 ± 0.020	0.105 ± 0.085	17.809 ± 0.418	–
EmBER–variant 3	0.322 ± 0.113	0.932 ± 0.001	0.106 ± 0.130	0.000 ± 0.000	–
EmBER–variant 4	0.258 ± 0.008	0.934 ± 0.001	0.028 ± 0.008	141.465 ± 0.622	–
EmBER–variant 5	0.758 ± 0.089	0.633 ± 0.009	0.313 ± 0.071	224.025 ± 3.412	–
EmBER–variant 6	0.673 ± 0.003	0.571 ± 0.003	0.022 ± 0.009	20.482 ± 0.565	–
EmBER–variant 7	0.687 ± 0.002	0.525 ± 0.044	0.045 ± 0.014	199.823 ± 1.191	–

Table 12: Sensitivity and ablation results for data dimension: $d = 50$.

Method	NLL ↓	Accuracy ↑	ECE ↓	Time (s)	PV (NLL)
Bayes-x	0.232 ± 0.001	0.935 ± 0.000	0.003 ± 0.001	3.345 ± 0.401	–
EmBER	$\mathbf{0.386 \pm 0.078}$	$\mathbf{0.882 \pm 0.015}$	0.108 ± 0.069	19.376 ± 1.562	–
ERM	0.734 ± 0.003	0.562 ± 0.006	0.122 ± 0.005	3.921 ± 0.036	0.021
IRM	0.735 ± 0.014	0.554 ± 0.005	0.118 ± 0.016	6.359 ± 0.714	0.021
V-REx	0.726 ± 0.008	0.553 ± 0.005	$\mathbf{0.108 \pm 0.007}$	5.380 ± 0.700	0.021
Fishr	0.744 ± 0.002	0.558 ± 0.005	0.132 ± 0.001	6.033 ± 0.757	0.021
EmBER–variant 2	0.671 ± 0.008	0.579 ± 0.008	0.057 ± 0.015	18.683 ± 1.345	–
EmBER–variant 3	0.310 ± 0.091	0.933 ± 0.002	0.103 ± 0.103	0.000 ± 0.000	–
EmBER–variant 4	0.277 ± 0.012	0.929 ± 0.005	0.045 ± 0.007	140.905 ± 1.172	–
EmBER–variant 5	0.760 ± 0.089	0.636 ± 0.001	0.305 ± 0.080	226.697 ± 2.373	–
EmBER–variant 6	0.686 ± 0.002	0.561 ± 0.003	0.039 ± 0.008	20.606 ± 0.696	–
EmBER–variant 7	0.688 ± 0.002	0.529 ± 0.052	0.050 ± 0.022	197.960 ± 3.629	–

Table 13: Sensitivity and ablation results for data dimension: $d = 100$.

Method	NLL ↓	Accuracy ↑	ECE ↓	Time (s)	PV (NLL)
Bayes-x	0.232 ± 0.002	0.934 ± 0.001	0.002 ± 0.001	3.055 ± 0.051	–
EmBER	$\mathbf{1.773 \pm 0.331}$	0.530 ± 0.008	$\mathbf{0.349 \pm 0.026}$	26.296 ± 0.713	–
ERM	2.169 ± 0.076	0.531 ± 0.007	0.378 ± 0.002	5.045 ± 0.509	0.213
IRM	2.117 ± 0.018	0.532 ± 0.005	0.377 ± 0.006	7.186 ± 0.666	0.213
V-REx	2.175 ± 0.051	0.532 ± 0.007	0.379 ± 0.008	5.240 ± 0.129	0.213
Fishr	2.144 ± 0.098	$\mathbf{0.532 \pm 0.007}$	0.378 ± 0.007	7.724 ± 0.294	0.213
EmBER–variant 2	1.209 ± 0.081	0.529 ± 0.003	0.279 ± 0.027	25.465 ± 0.879	–
EmBER–variant 3	1.167 ± 0.144	0.631 ± 0.019	0.240 ± 0.018	0.000 ± 0.000	–
EmBER–variant 4	1.702 ± 0.228	0.616 ± 0.019	0.298 ± 0.024	151.139 ± 0.521	–
EmBER–variant 5	0.748 ± 0.064	0.572 ± 0.005	0.130 ± 0.079	223.441 ± 2.586	–
EmBER–variant 6	1.970 ± 0.284	0.530 ± 0.006	0.353 ± 0.025	27.924 ± 0.794	–
EmBER–variant 7	0.688 ± 0.003	0.533 ± 0.059	0.056 ± 0.027	195.873 ± 1.272	–

Table 14: Sensitivity and ablation results for data dimension: $d = 1000$.

D.2 Additional Results for Main-Text Experiments

Tables 15–19 report the main-text results, augmented with ECE and running time. The times are recorded on a single CPU 5 performance cores and 6 efficiency cores, for all experiments except Colored MNIST where a single A100 GPU was used.

Method	NLL ↓	Accuracy ↑	ECE ↓	Time (s)	PV (NLL)
Bayes-x	0.233 ± 0.004	0.934 ± 0.002	0.002 ± 0.001	3.090 ± 0.379	–
EmBER	$\mathbf{0.319 \pm 0.069}$	$\mathbf{0.928 \pm 0.003}$	0.112 ± 0.078	18.920 ± 1.117	–
ERM	0.652 ± 0.005	0.577 ± 0.009	0.103 ± 0.013	4.373 ± 0.954	0.015
IRM	0.654 ± 0.006	0.573 ± 0.012	$\mathbf{0.086 \pm 0.022}$	6.053 ± 0.789	0.015
V-REx	0.640 ± 0.026	0.572 ± 0.016	0.109 ± 0.021	5.008 ± 0.658	0.015
Fishr	0.652 ± 0.006	0.582 ± 0.011	0.097 ± 0.010	5.812 ± 0.712	0.015

Table 15: Additional results for the parametric simulation with baseline values.

Method	NLL ↓	Accuracy ↑	ECE ↓	Time (s)	PV (NLL)
EmBER	$\mathbf{0.519 \pm 0.077}$	$\mathbf{0.787 \pm 0.071}$	$\mathbf{0.076 \pm 0.021}$	149.269 ± 37.363	–
ERM	0.863 ± 0.090	0.650 ± 0.021	0.194 ± 0.027	45.019 ± 1.007	0.004
IRM	0.572 ± 0.007	0.747 ± 0.017	0.101 ± 0.014	0.041 ± 0.006	0.203
V-REx	0.605 ± 0.019	0.684 ± 0.011	0.081 ± 0.011	70.164 ± 0.623	0.140
Fishr	0.917 ± 0.061	0.639 ± 0.020	0.207 ± 0.021	101.795 ± 1.436	0.004

Table 16: Additional results for the colored-MNIST simulation.

Method	NLL ↓	Accuracy ↑	ECE ↓	Time (s)	PV (NLL)
EmBER	$\mathbf{0.221 \pm 0.019}$	$\mathbf{0.921 \pm 0.007}$	$\mathbf{0.048 \pm 0.011}$	32.282 ± 0.690	–
ERM	0.390 ± 0.001	0.897 ± 0.001	0.104 ± 0.002	4.293 ± 1.548	< 0.001
Fishr	0.384 ± 0.004	0.894 ± 0.001	0.111 ± 0.002	6.666 ± 0.227	< 0.001
IRM	0.383 ± 0.009	0.891 ± 0.003	0.115 ± 0.006	6.781 ± 0.490	< 0.001
VREx	0.382 ± 0.009	0.892 ± 0.003	0.114 ± 0.005	5.154 ± 0.271	< 0.001

Table 17: Additional results for the quasar-star classification experiment.

Method	NLL ↓	Accuracy ↑	ECE ↓	Time (s)	PV (NLL)
EmBER	**0.850 ± 0.112**	**0.591 ± 0.027**	**0.131 ± 0.050**	1.163 ± 0.076	–
ERM	1.685 ± 0.136	0.587 ± 0.016	0.287 ± 0.018	0.242 ± 0.274	0.002
Fishr	1.449 ± 0.226	0.582 ± 0.021	0.255 ± 0.035	0.191 ± 0.016	0.008
IRM	1.548 ± 0.224	0.588 ± 0.011	0.265 ± 0.025	0.199 ± 0.015	0.007
VREx	0.903 ± 0.087	0.571 ± 0.014	0.157 ± 0.026	0.153 ± 0.009	0.543

Table 18: Additional results for the microbiome classification experiment.

Method	AUROC ↑	AUPRC ↑	ECE ↓	Time (s)	PV (AUROC)
EmBER	**0.740 ± 0.008**	**0.179 ± 0.008**	0.028 ± 0.009	43.754 ± 0.921	–
ERM	0.692 ± 0.017	0.153 ± 0.013	0.039 ± 0.004	5.718 ± 0.348	0.003
Fishr	0.697 ± 0.014	0.156 ± 0.008	0.039 ± 0.005	12.690 ± 0.425	0.002
IRM	0.724 ± 0.012	0.169 ± 0.012	**0.023 ± 0.004**	13.422 ± 0.671	0.112
VREx	0.697 ± 0.012	0.154 ± 0.011	0.029 ± 0.007	8.814 ± 0.165	0.001

Table 19: Additional results for the sepsis prediction experiment.

D.3 Method-Agreement Analysis

To better understand where EmBER differs from the competing methods, we decompose test examples by method agreement. In Figures 5 and 6 the left panels report the fraction of test examples on which all methods either succeed or fail. The right panels focus on disagreement cases where EmBER is correct, and report how often each competing method is also correct on those examples. In both datasets, no competing method succeeds consistently on the EmBER-success disagreement subset, indicating that these cases are not simply easy examples solved by all methods.

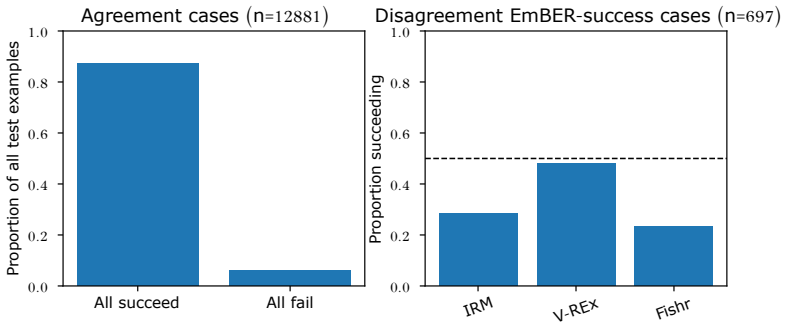


Figure 5: Agreement between methods in the quasar-star classification experiment. Left: proportion of test examples on which all methods are correct or all methods are incorrect. Right: among disagreement cases where EmBER is correct, the proportion also classified correctly by each competing method. The dashed line marks 0.5.

E Additional Experimental Details

E.1 Motivating Example

For Figure 3, we set $\alpha = 0.85$ and $\rho_y = 0.10$. Environment-specific prevalences were sampled as $\pi_e \sim \mathrm{Beta}(0.5, 0.5)$, so that $\mu = \mathbb{E}_{p(e)}[\pi_e] = 0.5$. For the first two panels, we sampled $m = 30$ observed environments and evaluated the conditional weights at $x \in \{0.15, 0.50, 0.85\}$. For the convergence panel, we used $m \in \{20, 50, 100, 200, 500, 1000, 2000, 5000\}$ observed environments and repeated the experiment 500 times. The uniform-weight limiting error was approximated by Monte Carlo integration using 200,000 draws of π_e. All curves were evaluated on an equally spaced grid of 500 values in $[0.001, 0.999]$.

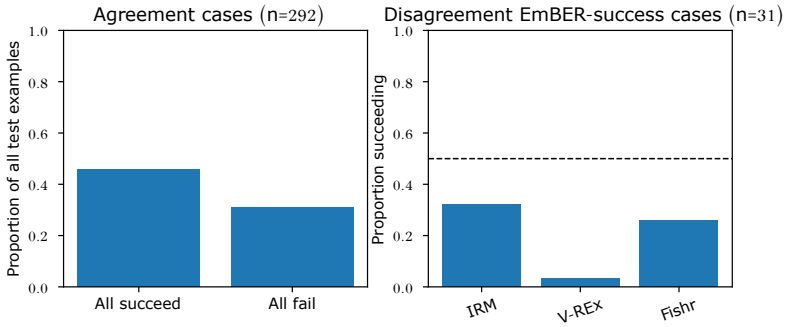


Figure 6: Agreement between methods in the microbiome classification experiment. Left: proportion of test examples on which all methods are correct or all methods are incorrect. Right: among disagreement cases where EmBER is correct, the proportion also classified correctly by each competing method. The dashed line marks 0.5.

E.2 Parametric Model

The simulation is designed so that environments affect the observed data only through the latent variable z. Each environment j is associated with a branch index b_j. We use branch values

$$b_j \in \{-3, -2, -1, 0, 1, 2\}. \tag{70}$$

For each environment, we draw a branch center

$$m_j = \Delta b_j + \phi_0 + \delta_j, \qquad \delta_j \sim \mathcal{N}(0, \sigma_\delta^2). \tag{71}$$

Given m_j, observations in that environment are generated independently as

$$z_{ij} \mid e_j \sim \mathcal{N}(m_j, \sigma_z^2). \tag{72}$$

The observed covariates are generated from z_{ij} by

$$x_{ij,1} = \cos(z_{ij}) + \varepsilon_{ij,1}, \tag{73}$$

$$x_{ij,2} = \sin(z_{ij}) + \varepsilon_{ij,2}, \tag{74}$$

$$x_{ij,3} = \rho z_{ij} + \varepsilon_{ij,3}, \tag{75}$$

and

$$x_{ij,k} = \varepsilon_{ij,k}, \qquad k = 4, \ldots, d. \tag{76}$$

The noise variables satisfy

$$\varepsilon_{ij,1}, \varepsilon_{ij,2} \sim \mathcal{N}(0, \sigma_{\text{phase}}^2), \tag{77}$$

$$\varepsilon_{ij,3} \sim \mathcal{N}(0, \sigma_{\text{branch}}^2), \tag{78}$$

and

$$\varepsilon_{ij,k} \sim \mathcal{N}(0, \sigma_{\text{noise}}^2), \qquad k = 4, \ldots, d. \tag{79}$$

The label is generated by

$$y_{ij} \mid z_{ij} \sim \text{Bernoulli}\left(\sigma\left(\beta_0 + \beta_1 \sin(z_{ij}/2)\right)\right), \tag{80}$$

where $\sigma(t) = (1 + \exp(-t))^{-1}$. Therefore,

$$p(e, z, x, y) = p(e)\, p(z \mid e)\, p(x \mid z)\, p(y \mid z) \tag{81}$$

as in the assumed model.

The key feature of this construction is that (x_1, x_2) reveal the phase of z but not its branch. For every integer k,

$$\cos(z + 2\pi k) = \cos(z), \qquad \sin(z + 2\pi k) = \sin(z). \tag{82}$$

However, adjacent branches have opposite label mechanisms because

$$\sin((z + 2\pi)/2) = -\sin(z/2). \tag{83}$$

Thus, $p(y \mid x, e)$ varies across environments, even though the latent mechanism $p(y \mid z)$ is stable.

The baseline parameters are

$$m_{\text{train}} = 6, \qquad n = 500, \qquad d = 10, \tag{84}$$

$$\Delta = 2\pi, \qquad \phi_0 = \pi/2, \qquad \sigma_\delta = 0.05, \tag{85}$$

and

$$\sigma_z = 0.35, \qquad \sigma_{\text{phase}} = 0.05, \qquad \rho = 0.1, \qquad \sigma_{\text{branch}} = 0.1, \qquad \sigma_{\text{noise}} = 1. \tag{86}$$

The label parameter is

$$\beta = 4. \tag{87}$$

The validation and test environments use the same number of observations per environment as the training environments.

Sensitivity analysis We study the effect of each simulation parameter separately. Starting from the baseline configuration, we vary one parameter at a time:

$$\rho \in \{0.1, 0.15, 0.2\}, \tag{88}$$

$$m_{\text{train}} \in \{6, 12, 36\}, \tag{89}$$

$$n \in \{100, 500, 1000\}, \tag{90}$$

$$d \in \{10, 50, 100, 1000\}. \tag{91}$$

Training, validation, and test environments For every configuration, we generate three independent collections of environments: training environments, validation environments, and test environments. The training set contains m_{train} environments. The validation set contains 12 new environments, and the test set contains 50 new environments. All validation and test environments are sampled from the same environment population as the training environments.

The validation environments are used only for selecting regularization coefficients for IRM, VREx, and Fishr over a grid search of $\lambda \in \{0.1, 1, 10\}$.

Models All methods receive only the observed covariates x, labels y, and training-environment labels e when required by the method. The latent variable z is used only to compute oracle reference performance. All methods use the same MLP with one hidden layer of width 16 and ReLU activations, followed by a 32-dimensional linear layer. The latent dimension of EmBER was set to 1.

For EmBER, we include the environment-classification term for estimating $\hat{p}(e \mid x)$. We evaluate two EmBER variants. The first is the full EmBER predictor, which uses learned environment weights $\hat{p}(e \mid x)$. The second is a uniform-weight ablation, which replaces the learned environment weights by uniform weights at prediction time.

Covariate optimal-Bayes oracle reference We report the Bayes-optimal predictor based on the observed covariates x:

$$p(y = 1 \mid x) = \int p(y = 1 \mid z) p(z \mid x) dz. \tag{92}$$

Here, $p(z \mid x)$ is not available in closed form since x_1 and x_2 depend on $\cos(z)$ and $\sin(z)$. We therefore compute the Bayes predictor numerically on a dense grid over z, using the true data-generating parameters.

Optimization and evaluation All methods are trained for 10 epochs using Adam with learning rate 10^{-3} and batch size 64. EmBER predictions are estimated with 1000 Monte Carlo samples. Expected calibration error is computed using 10 equally spaced probability bins.

E.3 Colored MNIST

For data generation we use $\alpha = \beta = 0.95$ and set

$$p(y=0|z=1) = p(y=0|z=2) = \alpha, \qquad p(y=1|z=3) = p(y=1|z=4) = \alpha,$$
$$p(c=\text{red}|z=1) = p(c=\text{red}|z=3) = \beta, \qquad p(c=\text{green}|z=2) = p(c=\text{green}|z=4) = \beta.$$

We set prototype-specific digit probabilities to

$$\begin{aligned}
p(d|z=1) &= (0.30, 0.22, 0.22, 0.10, 0.08, 0.06, 0.02, 0, 0, 0),\\
p(d|z=2) &= (0, 0, 0.02, 0.06, 0.08, 0.10, 0.22, 0.22, 0.22, 0.08),\\
p(d|z=3) &= (0.08, 0, 0, 0.08, 0.02, 0.22, 0.08, 0.22, 0.08, 0.22),\\
p(d|z=4) &= (0.08, 0.22, 0.22, 0.02, 0.22, 0, 0.10, 0, 0.14, 0).
\end{aligned}$$

We use six training environments with

$$p(z|e_1) = (0.49, 0.01, 0.01, 0.49), \quad p(z|e_2) = (0.45, 0.05, 0.05, 0.45), \quad p(z|e_3) = (0.40, 0.10, 0.10, 0.40),$$
$$p(z|e_4) = (0.30, 0.20, 0.20, 0.30), \quad p(z|e_5) = (0.20, 0.30, 0.30, 0.20), \quad p(z|e_6) = (0.10, 0.40, 0.40, 0.10).$$

These mixtures induce an association between color and label in the pooled training distribution: red images are predominantly assigned label 0, while green images are predominantly assigned label 1. Note that this association is not produced by an environment-specific coloring rule; rather, it emerges from the stable prototype mechanisms after marginalizing over the environment-dependent mixture $p(z|e)$.

For the test environment we set

$$p(z|e_{\text{test}}) = (0.01, 0.49, 0.49, 0.01).$$

Implementation details For each training environment we generate 5000 examples, and for the test environment we generate 10000 examples.

All methods use the same convolutional feature extractor. The feature extractor consists of two convolutional blocks, each with a 3×3 convolution, ReLU activation, and 2×2 max pooling, followed by a fully connected layer with ReLU activation. The resulting representation dimension is 32.

For our method, the encoder $g_\theta(x, y, e)$ takes the image, binary label, and environment index as input and outputs the mean and log-variance of a one-dimensional Gaussian variational distribution. The decoder $f_\phi^{(0)}$ maps a latent sample z to a binary logit for y. The auxiliary model $f_\varphi^{(1)}(x, e)$ maps an image and environment index to a binary logit estimating $p_\varphi(y = 1|x, e)$. The environment classifier $h_\psi(x)$ outputs a categorical distribution over the six training environments.

During training, for each minibatch observation (x_i, y_i, e_i), we evaluate $g_\theta(x_i, y_i, e)$ for all six training environments $e \in \mathcal{E}_{\text{tr}}$. We also evaluate $g_\theta(x_i, 0, e)$ and $g_\theta(x_i, 1, e)$ for all training environments in order to form

$$q_e(z|x_i) = p_\varphi(y=0|x_i, e)\, q_e(z|x_i, 0) + p_\varphi(y=1|x_i, e)\, q_e(z|x_i, 1). \tag{93}$$

The estimates $\hat{p}_\psi(e|x_i)$ are obtained by applying a softmax to the output of $h_\psi(x_i)$, and

$$\hat{p}(e|x_i, y_i) = \frac{\hat{p}_\psi(e|x_i)\, p_\varphi(y_i|x_i, e)}{\sum_{e' \in \mathcal{E}_{\text{tr}}} \hat{p}_\psi(e'|x_i)\, p_\varphi(y_i|x_i, e')}. \tag{94}$$

The posterior coupling term is estimated using one reparametrized sample from each $q_e(z|x_i, y_i)$. The prior coupling term is estimated using one sample from each mixture $q_e(z|x_i)$; this sample is obtained by first sampling the Bernoulli label component and then sampling from the selected Gaussian component.

All models are trained for 10 epochs with Adam, learning rate 10^{-3}, batch size 128, and gradient clipping at norm 5. Cross-validation over the grid $\{0.01, 0.1, 1, 10, 100\}$, using validation NLL as the selection criterion, yielded

$$\lambda_{\text{IRM}} = 10, \qquad \lambda_{\text{VREx}} = 10, \qquad \lambda_{\text{Fishr}} = 0.01. \tag{95}$$

At test time, for each test image x, we compute $\hat{p}_\psi(e|x)$ for all training environments, construct $q_e(z|x)$ for each of them, and approximate

$$\hat{p}(y=1|x) = \sum_{e \in \mathcal{E}_{\text{tr}}} \hat{p}_\psi(e|x) \int p_\phi(y=1|z)\, q_e(z|x)\, dz \tag{96}$$

by Monte Carlo. In the final evaluation we use 1000 samples from each Gaussian component. Accuracy is computed by thresholding $\hat{p}(y=1|x)$ at 0.5, and calibration is measured using 10-bin expected calibration error.

E.4 Quasar-Star Classification

We query spectroscopically confirmed objects from SDSS and retain only objects whose spectroscopic class is either `QSO` or `STAR`. The positive class corresponds to quasars and the negative class corresponds to stars.

We keep only objects with zero spectroscopic warning flag, clean photometry, primary photometric detections, and r-band magnitude between 14 and 22. We query at most $50{,}000$ objects before the subsequent class restriction to `QSO` and `STAR`. Rows with missing values in the retained variables are removed after feature construction.

Each object has right ascension and declination in the ICRS coordinate system. We convert these coordinates to Galactic coordinates and compute the absolute Galactic latitude $|b_i|$. We then partition the objects into 5 quintile bins of $|b_i|$. This construction induces environments corresponding to different sky regions, possibly leading to variation in stellar density and quasar prevalence. We select a shifted held-out environment according to this quasar prevalence.

From the ugriz magnitudes, we construct adjacent color features

$$u_i - g_i, \qquad g_i - r_i, \qquad r_i - i_i, \qquad i_i - z_i, \tag{97}$$

and additionally include the r-band magnitude.

We standardize the covariates using the training environments and apply the resulting transformation to the held-out environments.

Figure 7 reports train and test standardized feature distributions.

Implementation details All methods use the same two-layer feature extractor with width 16, ReLU activations and representation dimension 8.

We train all methods using minibatches of size 128 and Adam with learning rate 10^{-3}. Cross-validation over the grid $\{0.01, 0.1, 1, 10, 100\}$, using validation NLL as the selection criterion, yielded

$$\lambda_{\text{IRM}} = \lambda_{\text{VREx}} = \lambda_{\text{Fishr}} = 0.01. \tag{98}$$

For EmBER, held-out probabilities are estimated using Monte Carlo marginalization with 1000 samples.

E.5 Microbiome classification

We use the five colorectal-cancer cohorts available in the processed archive, namely `crc_baxter`, `crc_xiang`, `crc_zackular`, `crc_zeller`, and `crc_zhao`. For each cohort, we load the sample metadata and the corresponding OTU table from the RDP-annotated output files.

We exclude samples annotated as adenoma or polyp, so the task is restricted to cancer-versus-control classification. We preprocess the OTU tables by collapsing taxa to the genus level. For each sample, genus counts are converted to relative abundances, yielding one compositional feature vector per sample. We merge these features with the detected disease labels using the sample identifiers. After combining cohorts, missing genera are filled with zero abundance. We then retain only genera whose empirical prevalence across the combined dataset is at least 1%.

We remove cohorts that do not contain both classes, resulting in exclusion of `crc_zhao` which contains only controls. Evaluation is performed on one held-out cohort. We train on three cohorts, `crc_zeller`, `crc_xiang`, and `crc_zackular`, and evaluate on the held-out cohort `crc_baxter`.

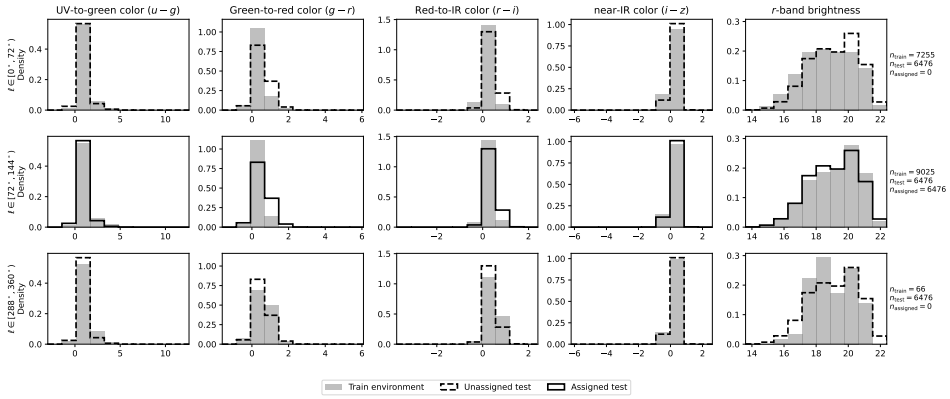


Figure 7: SDSS feature distributions by selected training environment. For all test points highest weight is assigned to environment $\ell \in [72^\circ, 144^\circ)$, whose feature distribution closely matches the test distribution.

Before fitting the models, genus-abundance features are standardized using the training environments only, and the same transformation is applied to the held-out test environment.

Implementation details All methods use the same linear feature map followed by a binary classifier, with representation dimension 16. EmBER latent dimension was set to 1. We compare EmBER to ERM, IRM, VREx, and Fishr. All models are trained for 50 epochs with Adam, learning rate 10^{-3}, and batch size 128. Cross-validation over the grid $\{0.01, 0.1, 1, 10, 100\}$, using validation NLL as the selection criterion, yielded

$$\lambda_{\text{IRM}} = 10, \qquad \lambda_{\text{V-REx}} = 100, \qquad \lambda_{\text{Fishr}} = 100. \tag{99}$$

For EmBER, held-out probabilities are estimated using Monte Carlo marginalization with 1000 samples.

For nearest-neighbor analysis, for each test observation x_i, we restrict the training data to the assigned environment $\hat{e}_i$ and find the nearest training observation

$$j(i) = \arg\min_{j: e_j = \hat{e}_i} \|x_j - x_i\|_2. \tag{100}$$

in the standardized feature space.

E.6 Sepsis prediction

We construct the sepsis experiment from the public PhysioNet 2019 Challenge training data. We use up to 15,000 patient files from each of the two public sources, `training_setA` and `training_setB`. Each file contains an hourly ICU time series for a single patient. We convert each file into one observation by summarizing the first 24 hours of the stay.

The covariates consist of demographic variables and summaries of early vital signs and laboratory measurements. For each time-varying measurement, we compute its last observed value, mean, minimum, maximum, and missingness fraction during the observation window.

We train on five source-unit environments and evaluate on a held-out source-unit environment. Specifically, the training environments are `A_unit2`, `B_unit1`, `A_unit1`, `B_unit2`, and `B_unknown`, and the test environment is `A_unknown`.

After this split, we filter features using only the training environments, retaining features with at most 5% missing values in the training data. Remaining missing values are imputed with the corresponding training-set median. We then standardize covariates using the training-set mean and standard deviation and apply the same transformation to the held-out test environment.

Implementation details All methods use the same representation network: a one-hidden-layer multilayer perceptron with hidden width 16 and output dimension 8. For EmBER, the latent dimension

was set to 1. We train all methods for 50 epochs with minibatch size 128 and learning rate 10^{-3}. Cross-validation over the grid $\{0.01, 0.1, 1, 10, 100\}$, using validation NLL as the selection criterion, yielded

$$\lambda_{\mathrm{IRM}} = 10, \qquad \lambda_{\mathrm{VREx}} = 10, \qquad \lambda_{\mathrm{Fishr}} = 0.1. \tag{101}$$

For EmBER, held-out probabilities are estimated using Monte Carlo marginalization with 1000 samples.