Context is Environment

Sharut Gupta*
MIT CSAIL
sharut@mit.edu

Stefanie Jegelka
MIT CSAIL
stefje@csail.mit.edu

David Lopez-Paz
Meta AI
dlp@meta.com

Kartik Ahuja
Meta AI
kartikahuja@meta.com

Abstract

Two lines of work are taking center stage in AI research. On the one hand, increasing efforts are being made to build models that generalize out-of-distribution (OOD). Unfortunately, a hard lesson so far is that no proposal convincingly outperforms a simple empirical risk minimization baseline. On the other hand, large language models (LLMs) have erupted as algorithms able to learn *in-context*, generalizing on-the-fly to the eclectic contextual circumstances. We argue that *context is environment*, and posit that in-context learning holds the key to better domain generalization. Via extensive theory and experiments, we show that paying attention to context—unlabeled examples as they arrive—allows our proposed In-Context Risk Minimization (ICRM) algorithm to *zoom-in* on the test environment risk minimizer, leading to significant OOD performance improvements. The take-home message from all this is two-fold: From all of this, two messages are worth taking home: researchers in domain generalization should consider *environment as context*, and harness the adaptive power of in-context learning. Researchers in LLMs should consider *context as environment*, to better structure data towards generalization.

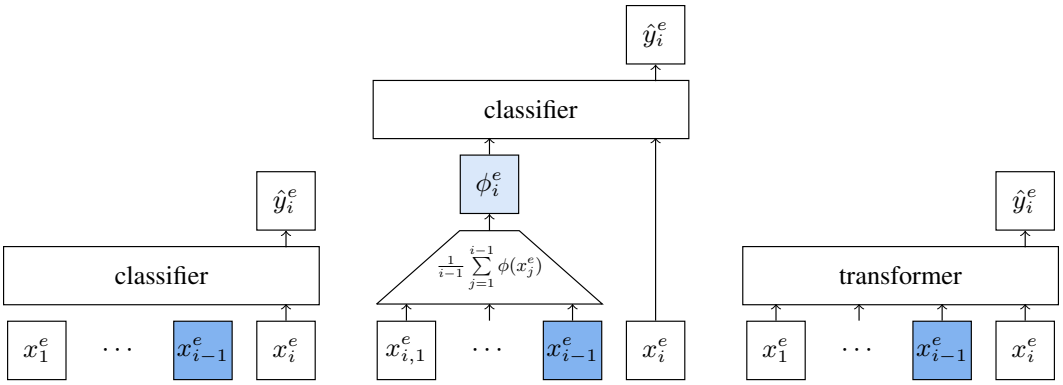
1 Introduction

Two predominant themes are emerging in AI research. On one hand, there is a mounting emphasis on building systems that generalize across a wide range of test environments. So far the bitter lesson is that no algorithm geared towards out-of-distribution (OOD) generalization convincingly outperforms a simple empirical risk minimization (ERM) baseline across standard real-world benchmarks [16, 14, 56]. On the other hand, large language models [33, 46, LLMs] are taking the world by storm. A standout feature of LLMs is their ability to learn *in-context*, enabling them to generalize on-the-fly to the eclectic user-driven prompts [8]. When interacting with LLMs, one feels closer towards solving the puzzle of OOD generalization. Could LLMs hold a key piece to this puzzle?

This paper suggests a positive answer, establishing a strong parallel between the concept of *environment* in domain generalization, and the concept of *context* in next-token prediction. On the one hand, describing *environments as context* opens the door to using powerful next-token predictors off-the-shelf, with their adaptability to learn in-context, to address domain generalization problems. This allows us to move from coarse domain indices to fine and compositional contextual descriptions, helpful to amortize learning across similar environments. On the other hand, using *context as environment* can help LLM researchers to use various domain generalization methods such as distributionally robust optimization [41, 54, DRO] across varying contexts.

*Most of the work done during an internship at Meta AI (FAIR), Paris.

R0-FoMo: Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models at NeurIPS 2023.



(a) Invariance DG. (b) Marginal transfer DG. (c) In-context DG (ours).

Figure 1: Three frameworks for domain generalization (DG), predicting the target y_i^e from the input x_i^e in test environment e. Depicted in blue, x_{i-1}^e contains relevant features for the current prediction. (a) Invariance DG discards all of the previously observed information from the test environment. (b) Marginal transfer DG summarizes all of the previously observed test inputs as a coarse embedding. (b) Our in-context DG directly observes all of the previous test inputs, allowing the search of "needle-in-the-haystack" signals, such as the relevant one, i.e., x_{i-1}^e.

Based on these insights, we propose a natural algorithm, *In-Context Risk Minimization (ICRM)*, illustrated in Figure 1c. Given examples (x_i^e, y_i^e) from environment e, we propose to address *out*-of-distribution prediction as *in*-distribution next-token prediction, training a machine:

$$y_i^e \approx h(x_i^e;\ \underbrace{x_1^e, \ldots, x_{i-1}^e}_{\text{environment} \approx \text{context}}\ ). \tag{1}$$

While the requested prediction y_i^e concerns only the input x_i^e, the machine can now pay attention to the test experience so far, extracting relevant environment information from instance and distributional features. Our theoretical results show that such in-context learners can utilize context to *zoom-in* on the empirical risk minimizer of the test environment, achieving competitive out-of-distribution performance. Further, we show that the extended input-context feature space in ICRM can reveal invariances that ERM-based algorithms ignore. Our extensive experiments demonstrate the efficacy of ICRM, and extensive ablations dissect and deepen our understanding of it.

2 The problem of domain generalization

The goal of domain generalization (DG) is to learn a predictor that performs well across a set of domains or environments $\mathcal{E}$ [31]. During training we have access to a collection of triplets $\mathcal{D} = \{(x_i, y_i, e_i)\}_{i=1}^n$. Each triplet contains a vector of features x_i, a target label y_i, and the index of the corresponding training environment $e_i \in \mathcal{E}_{\text{tr}} \subset \mathcal{E}$. Each example (x_i, y_i) is sampled independently from a joint distribution $P^e(X, Y)$. Using the dataset $\mathcal{D}$, we learn a predictor h that maps features to labels, while minimizing the worst risk across the set of all environments $\mathcal{E}$:

$$h^* = \arg\min_h \max_{e \in \mathcal{E}} R^e(h), \tag{2}$$

where $R^e(h) = \mathbb{E}_{(X,Y)\sim P^e}[\ell(h(X), Y)]$ is the risk of the predictor h in environment e, as measured by the expectation of the loss function ℓ with respect to the environment distribution P^e.

In broad strokes, domain generalization algorithms fall in one of the two following categories. In the first category, domain generalization algorithms based on invariance [31, 15, 35, 3], illustrated in Figure 1a, regularize predictors $h(x_i^e)$ to not contain any information about the environment e. This however results in removing a lot of signal about the prediction task. In the second category, domain generalization algorithms based on marginal transfer [7, 28, 57, 5] extract environment-specific information. These methods implement predictors $h(x_i^e, \phi_i^e)$, where $\phi_i^e = \frac{1}{i-1}\sum_{j=1}^{i-1}\phi(x_j^e)$ coarsely summarizes the environment e in terms of previously observed instances. Alas, all of these alternatives in the second category dilute relevant features found in individual examples. For example,

paradigm	training data	testing data	estimates
ERM	x, y	$x^{e'}$	$P(Y \mid X)$
IRM	x, y, e	$x^{e'}$	$P(Y \mid \phi^{\text{inv}}(X))$
LLM	z	z_t and context $z_{j<t}$	$P(Z_{t+1} \mid Z_t, \ldots, Z_1)$
ICRM	x, y, e	$x_t^{e'}$ and context $c_t^{e'} = (x_j^{e'})_{j<t}$	$P(Y \mid X, C) \rightsquigarrow P^{e'}(Y \mid X)$

Table 1: Different learning paradigms discussed in this work, together with their training data and testing data formats, as well as the estimated predictors. In our ICRM, we amortize the current input $x^{e'}$ and its context $c^{e'}$, containing previously experienced unlabeled examples from the same environment e', and "zoom-in" ($\rightsquigarrow$) to the appropriate local risk minimizer.

the size of the representation ϕ would have to grow linearly with the size of the training data to describe aspects corresponding to a small group of examples, such as extreme value statistics.

As a result, and despite all efforts, no proposal so far convincingly outperforms a simple empirical risk minimization baseline [47, ERM] across standard benchmarks [16, 14, 56]. Effectively, ERM simply pools all training data together and seeks the *global* empirical risk minimizer:

$$h^\dagger = \arg\min_h \sum_{e \in \mathcal{E}_{\text{tr}}} P(E = e) \cdot R^e(h). \tag{3}$$

Does the efficacy of ERM suggest that environmental information is useless? We argue that this is not the case. The key to our answer resides in a recently discovered emergent ability of next-token predictors, namely, in-context learning.

3 Next-token predictors and in-context learning

In next-token prediction, we aim to learn the conditional distribution

$$P(Z_{t+1} = z_{t+1} \mid Z_t = z_t, \ldots Z_1 = z_1), \tag{4}$$

describing the probability of observing the token z_{t+1} after having observed the sequence of tokens $(z_1, \ldots, z_t)$. The quintessential next-token prediction task is language modeling [6], where the sequence of tokens represents a snippet of natural language text. Most LLMs estimate Equation (4) via a transformer $z_{t+1} \approx h(z_t; z_{t-1}, \ldots, z_1)$ [48].

LLMs exhibit a certain ability, termed in-context learning (ICL), relevant to our interests. ICL is the ability to describe and learn about a learning problem from the sequence of tokens itself, called the context or prompt. To illustrate, consider the two following sequences:

$$\underbrace{\text{"You are talking to a teenager.}}_{\text{context } c_1} \; \underbrace{\text{Write a poem on gravitational fields."}}_{x_1}$$

$$\underbrace{\text{"You are talking to a Physics graduate.}}_{\text{context } c_2} \; \underbrace{\text{Write a poem on gravitational fields."}}_{x_2}$$

LLMs provide distinct responses, say y_1 and y_2 to these two sequences, adapting to the perceived audience. Although the model offers high-likelihood continuations to the prompts, it showcases compositional generalization by giving varied yet accurate answers based on context c_1 and c_2.

Since we train the machine to produce an enormous amount of completions, some of which start with partially overlapping contexts, the predictor has the opportunity to amortize learning to a significant degree i.e. use the trained model to generalize across unseen distributions rather than explicitly optimizing a separate model for each distribution. This is the desired ability to generalize over environments described in the previous section, which remained out of reach when using coarse domain indices.

4 Adaptive domain generalization via in-context learning

Our exposition has so far laid out two threads. First, Section 2 motivated the need for domain generalization algorithms capable of extracting relevant environment-specific features, at both the

example and distributional levels. To this end, we have argued to move away from coarse environment indices, and towards rich and amortizable descriptions shared in new circumstances. Second, Section 3 suggests understanding *context* as an opportunity to describe *environments* in precisely this manner. We now knit these threads together with a protocol to address domain generalization with in-context learners.

In-Context Risk Minimization (**ICRM**, Figure 1c):

- Collect a dataset of triplets $\mathcal{D} = \{(x_i, y_i, e_i)\}_{i=1}^n$ as described in Section 2. Initialize a next-token predictor $\hat{y} = h(x; c)$, tasked with predicting the label y associated to the input x, as supported by the context c.
- During training, select $e \in \mathcal{E}_{\text{tr}}$ at random. Draw t examples from this environment at random, construct one input sequence $(x_1^e, \ldots, x_t^e)$ and its associated target sequence $(y_1^e, \ldots, y_t^e)$. Update the next-token predictor to minimize the auto-regressive loss $\sum_{j=1}^t \ell(h(x_j^e; c_j^e), y_j^e)$, where the context is $c_j^e = (x_1^e, \ldots, x_{j-1}^e)$, for all $j = 2, \ldots, t$, and $c_1^e = \emptyset$.
- During test time, a sequence of inputs $(x_1^{e'}, \ldots, x_{t'}^{e'})$ arrives for prediction, one by one, all from the test environment $e' \in \mathcal{E}_{\text{te}}$. We predict $\hat{y}_j^{e'} = h(x_j^{e'}, c_j^{e'})$ for $x_j^{e'}$, where the context $c_j^{e'} = (x_1^{e'}, \ldots, x_{j-1}^{e'})$, for all $j = 2, \ldots, t'$, and $c_1^{e'} = \emptyset$.

A few critical remarks about the above proposal are in order. The most natural way to construct contexts is to use past samples that appear in the natural order in which data was collected. Since existing datasets do not provide such a refined ordering, we build contexts using environment indices that are more readily available. The proposal also requires the data at test time to be sampled from the same or slowly changing environments. Next, we develop theoretical guarantees on the behavior of ICRM. The results below concern the joint distribution of $\big((X_1, \cdots X_t), (Y_1, \ldots, Y_t), E\big)$, where each X_j, Y_j is an independent draw from environment E with distribution $P^E(X, Y)$. For query X_j, the context preceding it is $C_j = (X_1, \cdots, X_{j-1})$ and the environment underlying this context is E. To orient ourselves around these results, we recall three predictors featured in the exposition so far. First, the global empirical risk minimizer over the pooled training data, denoted by $h^\dagger$ in Equation (3), estimates $P(Y \mid X)$. Second, the environment risk minimizer estimates $P(Y \mid X, E)$. Third, our in-context risk minimizer estimates the conditional expectation $P(Y \mid X, C)$, denoted by

$$\tilde{h} = \arg\min_h \sum_{j=1}^t \mathbb{E}_{(X_j, C_j, Y_j)}[\ell(h(X_j; C_j), Y_j)]. \tag{5}$$

The sequel focuses on the binary cross-entropy loss ℓ. Our first result shows that, in the absence of context, ICRM *zooms-out* to behave conservatively.

Proposition 1 (Zoom-out)**.** *In the absence of context,* ICRM *behaves as the global empirical risk minimizer across the support of the training environments, i.e.,* $\tilde{h}(\cdot\,;\,\emptyset) = h^\dagger(\cdot)$.

The above result is built on the insight that ICRM is Bayes optimal at all context lengths and ERM is Bayes optimal for context $c = \emptyset$. Having established the connection between ICRM and ERM in the absence of any context, we now study the benefits of ICRM in the presence of sufficiently long contexts. The following result shows that, when provided with context from a training environment $e \in \mathcal{E}_{\text{tr}}$, our ICRM *zooms-in* and behaves like the appropriate environment risk minimizer, as shown in Table 1. We assume that $P(Y = 1 \mid X = x, E = e)$ is parametrized by a function $h^\star(x, \theta_x^e)$, where θ_x^e describes features of the environment relevant to the query x, for all $e \in \mathcal{E}$. We also assume there exists an ideal *amortization function* b that takes as input the query X and context C_t preceding it—both sampled from environment E—and approximates θ_X^E. Formally, the sequence of random variables $b(X, C_t)$ indexed by t converges almost surely to the random variable θ_X^E.

Theorem 1 (Full iid zoom-in)**.** *Let* $h^\star(x, \theta_x^e)$ *describe* $P(Y = 1 \mid X = x, E = e)$ *for all* $e \in \mathcal{E}$. *Further, we assume the existence of an* amortization function $b(X, C_t) \overset{a.s.}{\to} \theta_X^E$. *Then,* ICRM *zooms-in on the environment risk minimizer and achieves a cross-entropy loss over the training distribution*

$$\lim_{t \to \infty} H(Y \mid X, C_t) = H(Y \mid X, E).$$

Further, if $I(Y; E \mid X) > 0$, ICRM *has better performance than the global risk minimizer.*

Theorem 1 states that ICRM converges to empirical risk minimizer of the environment under infinitely long contexts. Next, we show that ICRM can partially zoom-in on the appropriate environment risk minimizer even with contexts of length of one.

Theorem 2 (Partial iid zoom-in)**.** *Suppose the joint distribution* $((X_1, \cdots X_t), (Y_1, \ldots, Y_t), E)$ *is Markov with respect to a Bayesian network. The query* X *and the environment* E *are statistically dependent and form the Markov blanket of* Y*. Then* ICRM *partially zooms-in on the environment risk minimizer, improving over the performance of the global empirical risk minimizer in terms of the cross-entropy loss. Further, the improvement is strictly monotonic in context length* t*.*

Next, we move to the out-of-distribution setting where the test environments can be different from the training environments. To provide theory for a domain generalization result, we must place some assumptions on the data generation process. In particular, and for all $e \in \mathcal{E}$, let

$$z \mid y, e \sim \mathcal{N}(\mu_e^y, \Sigma_e^y), \text{ and } x \leftarrow g(z), \tag{6}$$

where the latent variables z are sampled conditional on the label y and environment e from a Gaussian distribution with mean and covariance depending on (y, e), and are then mixed by a map g to generate the observations x. We summarize the environment in terms of the parameter vector $\gamma_e = \left[(p_e^y, \mu_e^y, \Sigma_e^y)_{y\in\{0,1\}}\right]$, where p_e^y is the probability of label y in environment e. Our next result shows that ICL algorithms that learn $h(x; c)$ exhibit robust behavior under distribution shifts. In contrast, such guarantees are not known for algorithms that generate predictors of the form $h(x)$.

Define δ_e to be a permutation of γ_e that swaps its two components. We construct the Voronoi cells corresponding to the points in the union of sets $\{\gamma_e\}_{e\in\mathcal{E}_{tr}}$ and $\{\delta_e\}_{e\in\mathcal{E}_{tr}}$. The set of points in the Voronoi cells corresponding to the set of points $\{\gamma_e\}_{e\in\mathcal{E}_{tr}}$ define the *Voronoi cells of the training environments*. Next, we show that ICL can perform in novel test environments sufficiently far away from the training environments, so long as they are in the Voronoi cells of training environments.

Theorem 3 (Full OOD zoom-in)**.** *Consider data triplets* (x, y, e) *generated from* $z \sim \mathcal{N}(\mu_e^y, \Sigma_e^y)$ *and* $x \leftarrow g(z)$*,* $\forall e \in \mathcal{E}$*, where* g *is the identity map (see Appendix A for extension to general diffeomorphism* g*). There exists an ICL algorithm that in the limit of infinitely long contexts produces Bayes optimal predictions for all the test environments in the Voronoi cells of the training environments.*

5 ICRM under the lens of invariance

Common advice in domain generalization recommends following the *invariance principle* to learn robust predictors [35, 3]. At first sight, one could argue that the proposed ICRM does not adhere to such a principle, as it is adapting to environment-specific information provided in the form of context. As we shall now illustrate, ICRM's implementation of ERM on the extended input-context feature space reveals invariant predictors that a vanilla implementation of ERM on the standard feature space fails to find. To see this, consider a linear least-squares regression problem mapping two-dimensional inputs $x = (x^1, x^2)$ into a target y under environments $e \in \mathcal{E}$ as $y = \alpha \cdot x^1 + \beta \cdot \mu_e^2 + \varepsilon$ where $\mu_e^i = \mathbb{E}[X^i \mid E = e]$, the pair (α, β) are invariant regression coefficients, and ε is an independent noise term. We make one simplifying assumption for pedagogic purposes. During training, we provide ICRM training directly with the relevant extended feature space $(x^1, x^2, \mu_e^1, \mu_e^2)$, instead of requiring the algorithm to learn such representation from general-form sequential context.

In this setup, ICRM learns to predict using $\alpha \cdot x^1 + 0 \cdot x^2 + 0 \cdot \mu_e^1 + \beta \cdot \mu_e^2$. In contrast, ERM trains a linear model on (x^1, x^2) and predicts using $\tilde{\alpha} \cdot x^1 + \tilde{\beta} \cdot x^2$. The main point is: if $\beta \neq 0$ and $\text{cov}(X^1, X^2) \neq 0$, then $\tilde{\alpha} \neq \alpha$, and the error of ERM in a new environment grows with the variance of x^1. On the other hand, ICRM estimates the true invariant coefficient α, and the resulting error is independent of variance of x^1, even in the absence of context during test time. For a derivation and generalization of these claims, see Appendix A.

6 Experiments

We experiment on how ICRM fares against competitive DG algorithms for different context sizes. We compare ICRM against marginal transfer methods such as Adaptive Risk Minimization [57, ARM], and test-time adaptation proposals such as TENT [52]. As a strong baseline, we also include ERM in our experimental protocol. We use a standardized neural network backbone (ConvNet or

ResNet-50, depending on the dataset) as described in Appendix C.4 to ensure a fair comparison across different algorithms. For ICRM, the same backbone is used to featurize the input, which is then processed by the decoder-only GPT-2 [36]. For fair comparisons, we adhere to DomainBed's protocols for training, hyper-parameter tuning, and testing [16]. We describe our experimental setup in detail in Appendix C.4. We assess these methods across four image classification benchmarks, each offering a unique problem setting. FEMNIST [11] contains MNIST digits and handwritten letters from individual writers as environments. Rotated MNIST concerns varied rotational angles as environments. Tiny ImageNet-C [17] introduces diverse image corruptions to create multiple environments. WILDS Camelyon17 [23] studies tumor detection and sourcing data from multiple hospitals as distinct environments. More details are provided in Appendix C.3.

Table 2: Average/worst OOD test accuracy for different context lengths, for Adaptive Risk Minimization (ARM), Empirical Risk Minimization (ERM), Test Entropy Minimization (TENT) and our ICRM on FEMNIST, Rotated MNIST, WILDS Camelyon17 and Tiny-ImageNet-C.

Data / method	**Average test accuracy**					**Worst case test accuracy**				
FEMNIST	0	25	50	75	100	0	25	50	75	100
ARM	49.5	83.9	84.4	84.7	84.6	23.6	59.5	60.7	57.0	58.8
TENT	78.1	77.9	81.2	82.5	83.3	55.2	57.2	63.3	65.9	67.2
ERM	**79.3**	79.3	79.3	79.3	79.3	59.0	59.0	59.0	59.0	59.0
ICRM	78.7	**87.2**	**87.4**	**87.5**	**87.8**	**59.8**	**69.3**	**70.6**	**70.6**	**70.6**
Rotated MNIST	0	25	50	75	100	0	25	50	75	100
ARM	36.5	94.2	95.1	95.3	95.5	28.2	85.3	87.2	87.9	87.9
TENT	94.1	88.0	91.9	93.8	94.3	80.2	88.5	88.5	80.2	81.3
ERM	**94.2**	94.2	94.2	94.2	94.2	80.8	80.8	80.8	80.8	80.8
ICRM	93.6	**96.1**	**96.2**	**96.2**	**96.2**	**82.5**	**88.5**	**88.5**	**88.8**	**88.8**
WILDS Camelyon17	0	25	50	75	100	0	25	50	75	100
ARM	61.2	59.5	59.7	59.7	59.7	same as average accuracy				
TENT	67.9	81.8	87.2	89.4	89.4					
ERM	68.6	68.6	68.6	68.6	68.6					
ICRM	**92.0**	**90.7**	**90.8**	**90.8**	**90.8**					
Tiny ImageNet-C	0	25	50	75	100	0	25	50	75	100
ARM	30.8	31.0	31.0	31.0	31.0	8.2	8.3	8.2	8.3	8.2
TENT	31.7	1.6	1.7	2.0	2.1	9.4	1.2	1.4	1.6	1.6
ERM	31.8	31.8	31.8	31.8	31.8	9.5	9.5	9.5	9.5	9.5
ICRM	**38.3**	**39.2**	**39.2**	**39.2**	**39.2**	**18.8**	**19.2**	**19.5**	**19.5**	**19.4**

We evaluate the performance of different approaches to distribution shifts for test context lengths of 0, 25, 50, 75, and 100 samples. We report an average across three independent runs of the entire sweep and its corresponding standard error, where we select the model with hyper-parameters corresponding to the highest validation accuracy. As shown in Table 2, ICRM outperforms all methods across context lengths over both the worst group and average accuracy, except at null context length on MNIST datasets, where ERM exceeds by 1%. Figure 4 zooms into the model's performance between no-context and 25 context samples, highlighting the consistent superiority of ICRM even with small contexts. Additionally, ICRM demonstrates gains in performance even in the absence of test context. Specifically for both WILDS Camelyon17 and Tiny ImageNet-C, ICRM outperforms baselines despite not leveraging any context from the test environment. This is because ICRM training still benefits from contexts as to find contextual features that ERM ignores.

Table 8 and Table 9 studies ICRM when samples in the training sequences are drawn i.i.d across environments. Further, Table 10 and Table 11 investigate the impact of bigger architecture on ARM and ERM. Lastly, Figure 5 visualizes the attention scores between examples in an input sequence and a novel input query, highlighting ICRM's ability to amortize learning.

7 Discussion

We introduced In-Context Risk Minimization (ICRM), a framework to address domain generalization as next-token prediction. ICRM learns in-context about environmental features by paying attention to unlabeled instances as they arrive, enabling competitive out-of-distribution generalization. While

prior work on DG focused on information removal as a guide to generalization, ICRM suggests that extending the feature space with the relevant environment information affords further invariance.

In the future, we would like to further understand how next-token prediction and in-context learning serve as a powerful mechanism to amortize and dynamically navigate trade-offs such as efficiency-resiliency, exploration-exploitation, specialization-generalization, and focusing-diversifying.

Acknowledgements

SG and SJ acknowledge funding from the Office of Naval Research grant N00014-20-1-2023 (MURI ML-SCOPE) and NSF award CCF-2112665 (TILOS AI Institute). We are thankful to Martin Arjovsky, Léon Bottou, Elvis Dohmatob, Badr Youbi Idrissi, Maxime Oquab, and Ahmed Touati for their valuable feedback and help.

References

[1] Kartik Ahuja and David Lopez-Paz. A closer look at in-context learning under distribution shifts. *arXiv*, 2023.

[2] Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *ICML*, 2020.

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

[4] Robert B Ash and Catherine A Doléans-Dade. *Probability and measure theory*. 2000.

[5] Yujia Bao and Theofanis Karaletsos. Contextual Vision Transformers for Robust Representation Learning. *arXiv*, 2023.

[6] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *NeurIPS*, 2000.

[7] Gilles Blanchard, Aniket Anand Deshmukh, Ürun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *JMLR*, 2011.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020.

[9] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi S Jaakkola. Invariant rationalization. In *ICML*, 2020.

[10] Yining Chen, Elan Rosenfeld, Mark Sellke, Tengyu Ma, and Andrej Risteski. Iterative feature matching: Toward provable domain generalization with logarithmic environments. *NeurIPS*, 2022.

[11] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *IJCNN*, 2017.

[12] Cian Eastwood, Alexander Robey, Shashank Singh, Julius Von Kügelgen, Hamed Hassani, George J Pappas, and Bernhard Schölkopf. Probable domain generalization via quantile risk minimization. *NeurIPS*, 2022.

[13] Cian Eastwood, Shashank Singh, Andrei Liviu Nicolicioiu, Marin Vlastelica, Julius von Kügelgen, and Bernhard Schölkopf. Spuriosity didn't kill the classifier: Using invariant predictions to harness spurious features. *arXiv preprint arXiv:2307.09933*, 2023.

[14] Jean-Christophe Gagnon-Audet, Kartik Ahuja, Mohammad Javad Darvishi Bayazi, Pooneh Mousavi, Guillaume Dumas, and Irina Rish. WOODS: Benchmarks for out-of-distribution generalization in time series. *TMLR*, 2023.

[15] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

[16] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv*, 2020.

[17] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv*, 2019.

[18] Pavel Izmailov, Polina Kirichenko, Nate Gruver, and Andrew G Wilson. On feature learning in the presence of spurious correlations. *NeurIPS*, 2022.

[19] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Enforcing predictive invariance across structured biomedical domains, 2020.

[20] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *AISTATS*, 2020.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2014.

[22] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv*, 2022.

[23] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.

[24] Masanori Koyama and Shoichiro Yamaguchi. Out-of-distribution generalization with maximal invariant predictor. *arXiv*, 2020.

[25] 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. *arXiv*, 2020.

[26] Sébastien Lachapelle, Pau Rodriguez, Yash Sharma, Katie E Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ica. In *Conference on Causal Learning and Reasoning*, 2022.

[27] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.

[28] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv*, 2016.

[29] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. *arXiv*, 2020.

[30] Maggie Makar, Ben Packer, Dan Moldovan, Davis Blalock, Yoni Halpern, and Alexander D'Amour. Causally motivated shortcut removal using auxiliary labels. In *AISTATS*, 2022.

[31] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, 2013.

[32] Jens Müller, Robert Schmier, Lynton Ardizzone, Carsten Rother, and Ullrich Köthe. Learning robust models using the principle of independent causal mechanisms. *arXiv*, 2020.

[33] OpenAI. GPT-4 Technical Report. *arXiv*, 2023.

[34] Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. In *ICLR*, 2021.

[35] 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*, 2016.

[36] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

[37] Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. In *ICML*, 2022.

[38] Alexander Robey, George J Pappas, and Hamed Hassani. Model-based domain generalization. *arXiv*, 2021.

[39] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *JMLR*, 2018.

[40] Walter Rudin. *Principles of mathematical analysis*. 1953.

[41] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv*, 2019.

[42] Olawale Salaudeen and Oluwasanmi Koyejo. Target conditioned representation independence (tcri); from domain-invariant to domain-general representations. *arXiv preprint arXiv:2212.11342*, 2022.

[43] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv*, 2021.

[44] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV Workshops*, 2016.

[45] Damien Teney, Ehsan Abbasnejad, and Anton van den Hengel. Unshuffling data for improved generalization. *arXiv*, 2020.

[46] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[47] Vladimir Vapnik. *Statistical learning theory*. Wiley, 1998.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[49] Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations in text classification. *NeurIPS*, 2021.

[50] Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. On calibration and out-of-domain generalization. *NeurIPS*, 2021.

[51] Yoav Wald, Gal Yona, Uri Shalit, and Yair Carmon. Malign overfitting: Interpolation can provably preclude invariance. *arXiv preprint arXiv:2211.15724*, 2022.

[52] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv*, 2020.

[53] Haoxiang Wang, Haozhe Si, Bo Li, and Han Zhao. Provable domain generalization via invariant-feature subspace recovery. In *ICML*, 2022.

[54] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *arXiv*, 2023.

[55] Sidney J Yakowitz and John D Spragins. On the identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 1968.

[56] Huaxiu Yao, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei W Koh, and Chelsea Finn. Wild-time: A benchmark of in-the-wild distribution shift over time. *NeurIPS*, 2022.

[57] Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: Learning to adapt to domain shift. *NeurIPS*, 2020.

[58] Yihua Zhang, Pranay Sharma, Parikshit Ram, Mingyi Hong, Kush Varshney, and Sijia Liu. What is missing in IRM training and evaluation? challenges and solutions. *arXiv*, 2023.

Appendix

A Theorems and Proofs

A.1 Proof of Proposition 1

Lemma 1. ICRM *is Bayes optimal at all context lengths.* *Suppose ℓ is the binary cross-entropy loss and the labels Y are binary. The optimal in-context learner $\tilde{h}$ (equation 5) satisfies the following condition, i.e., for each $k \in [t]$*

$$\tilde{h}(x_k; c_k) = \mathbb{E}[Y | X_k = x_k, C_k = c_k], \tag{7}$$

for almost all (c_k, x_k) in the support of training distribution except over a set of a measure zero, and where the expectation is over Y conditional on $[c_k, x_k]$. In other words, the in-context learner is Bayes optimal at each context length.

Proof. In this result, we consider the problem of binary classification. Suppose $h(x_k; c_k)$ is the predicted probability of class $Y = 1$ conditional on x_k and c_k. Define $\bar{h}(x_k; c_k) = \big[h(x_k; c_k), 1 - h(x_k; c_k)\big]$ describing the probability of both the classes.

From equation 5, recall that the objective of ICRM is to minimize

$$\sum_{j=1}^{t} \mathbb{E}_{(X_j, C_j, Y_j)}[\ell(h(X_j; C_j), Y_j)]. \tag{8}$$

Consider one of the terms in the sum above - $\mathbb{E}\big[\ell(h(X_k; C_k), Y_k)\big]$. Substituting ℓ as the cross-entropy in this term, we obtain

$$\mathbb{E}\big[\ell(h(X_k; C_k), Y_k)\big] = H(Y_k|X_k, C_k) + \mathbb{E}\big[\mathsf{KL}\big(P(Y_k|X_k, C_k)\big\|\bar{h}(X_k; C_k)\big)\big].$$

If $\bar{h}(X_k; C_k) = P(Y_k|X_k, C_k)$, then the second term in the above is zero and $\mathbb{E}\big[\ell(h(X_k; C_k), Y_k)\big]$ equals $H(Y_k|X_k; C_k)$. Since KL divergence is always non-negative, $H(Y_k|X_k, C_k)$ corresponds to the lowest value that can be achieved by $\mathbb{E}\big[\ell(h(X_k; C_k), Y_k)\big]$. If $\bar{h}(X_k; C_k) = P(Y_k|X_k, C_k)$ for all $k \in [t]$, then each of the terms in the sum in equation 8 are minimized. As a result, $\bar{h}(X_k; C_k) = P(Y_k|X_k, C_k)$ for all $k \in [t]$ is a solution to equation 5.

Consider another minimizer $h^{'}$ of equation 5 and define the corresponding distribution $\bar{h}^{'}$. For each $k \in [t]$, the second term $\mathbb{E}\big[\mathsf{KL}(P(Y_k|X_k, C_k)\|\bar{h}^{'}(X_k; C_k)\big]$ has to be zero for $\bar{h}^{'}$ to be a minimizer.

If $\mathbb{E}\big[\mathsf{KL}(P(Y_k|X_k, C_k)\|\bar{h}^{'}(X_k; C_k)\big] = 0$, then we claim that $\bar{h}^{'}(x_k; c_k) = P(Y_k|X_k = x_k, C_k = c_k)$ for almost all (x_k, c_k) in the support of training distribution except over a set of measure zero. If the probability measure associated with X_k, C_k is absolutely continuous w.r.t Lebesgue measure, then this claim follows from Theorem 1.6.6 [4]. If the probability measure associated with X_k, C_k is absolutely continuous w.r.t counting measure, then this claim trivially follows. □

We proved the above result for classification and cross-entropy loss for measures over X, C that are either absolutely continuous w.r.t Lebesgue measure or the counting measure. It is easy to extend the above result for regressions and least square loss; see Lemma 1 in [1].

Proposition 1 (Zoom-out)**.** *In the absence of context,* ICRM *behaves as the global empirical risk minimizer across the support of the training environments, i.e.,* $\tilde{h}(\cdot\,;\,\emptyset) = h^{\dagger}(\cdot)$.

Proof. From Lemma 1, it follows that $\tilde{h}(x_k; c_k) = \mathbb{E}[Y|X_k = x_k, C_k = c_k]$. The solution to empirical risk minimization is $h^{\dagger}(x) = \mathbb{E}[Y|X_1 = x]$, where the expectation is computed over the training distribution of Y conditional on x. When the context is empty, then we have $\tilde{h}(x; \emptyset) = \mathbb{E}[Y|X_1 = x] = h^{\dagger}(x)$ for almost all x in the support of training distribution except over a set of measure zero. □

A.2 Proof of Theorem 1

Before stating the proof of Theorem 1, we provide an example of an ideal amortization map $b(\cdot)$.

Example of ideal amortization map. Consider the example from equation **??**, where $y = \alpha \cdot x^1 + \beta \cdot \mu_e^2 + \varepsilon$. $P(Y = y|X = x, E = e) = p_\varepsilon(y - \alpha x^1 - \beta\mu_e^2)$, where p_ε is the probability density of noise. Observe that $P(Y = y|X = x, E = e)$ is parametrized in terms of μ_e^2 and the sequence of random variables $b(X, C_t) = \frac{1}{t-1}\sum_{j=1}^{t-1} X_j^2$ converge almost surely to μ_e^2, where X_j^2 is the second component of X_j and $C_t = (X_1, \cdots, X_{t-1})$.

For ease of exposition, we start with the case when all the concerned random variables $X, Y, C_t, E, b(X, C_t)$, where X is the current query and Y is its label and C_t is the context preceeding it sampled from environment E, and $b(\cdot)$ is the ideal amortization map, are discrete-valued with a finite support. Subsequently, we study more general settings.

Theorem 1 (Full iid zoom-in)**.** *Let $h^\star(x, \theta^e_x)$ describe $P(Y = 1 \mid X = x, E = e)$ for all $e \in \mathcal{E}$. Further, we assume the existence of an* amortization function $b(X, C_t) \overset{a.s.}{\to} \theta^E_X$*. Then,* ICRM *zooms-in on the environment risk minimizer and achieves a cross-entropy loss over the training distribution*

$$\lim_{t\to\infty} H(Y \mid X, C_t) = H(Y \mid X, E).$$

Further, if $I(Y; E \mid X) > 0$, ICRM *has better performance than the global risk minimizer.*

Proof. As stated above, in this proof, we work with discrete-valued $X, Y, C_t, E, b(X, C_t)$ that also have finite support, where X is the current query and Y is its label and C_t is the context preceeding it sampled from environment E, and $b(\cdot)$ is the ideal amortization map. Subsequently, we study more general settings.

Since each (X_j, Y_j) is sampled independently given a training environment E, we can conclude $I(Y; C_t|X, E) = 0$. Therefore,

$$I(Y; C_t|X, E) = 0 \implies H(Y|X, E) = H(Y|X, E, C_t).$$

Observe that for all $t \in \mathbb{Z}_+$

$$H(Y|X, E) = H(Y|X, E, C_t) \leq H(Y|X, C_t) \leq H(Y|X, b(X, C_t)), \tag{9}$$

where $\mathbb{Z}_+$ is the set of all positive integers. The first inequality in the above follows from the fact that conditioning reduces entropy. For the second inequality, we use the following property. Consider U, V as two random variables and define $W = a(V)$. Observe that $I(U; W|V) = 0 \implies H(U|V) = H(U|V, W) \leq H(U|W)$.

Since the inequality above equation 9 holds for all t, we obtain

$$H(Y|X, E) \leq \lim_{t\to\infty} H(Y|X, C_t) \leq \lim_{t\to\infty} H(Y|X, b(X, C_t)). \tag{10}$$

In the above, we use the following property. If $a_n \leq b_n, \forall n \in \mathbb{Z}_+$ and $\lim_{n\to\infty} a_n$ and $\lim_{n\to\infty} b_n$ exist, then $\lim_{n\to\infty} a_n \leq \lim_{n\to\infty} b_n$. In what follows, we will show that both the limits $\lim_{t\to\infty} H(Y|X, C_t)$ and $\lim_{t\to\infty} H(Y|X, b(X, C_t))$ exist. First observe that $H(Y|X, C_{t+1}) \leq H(Y|X, C_t)$ for all t as a result the sequence is decreasing bounded below by 0 and thus from monotone convergence theorem [40] $\lim_{t\to\infty} H(Y|X, C_t)$ exists. Next, we will show that $\lim_{t\to\infty} H(Y|X, b(X, C_t)) = H(Y|X, E)$. We will then combine it equation 10 to obtain what we intend to prove, i.e., $\lim_{t\to\infty} H(Y|X, C_t) = H(Y|X, E)$.

For each $X = x$ and $E = e$ in the support of training distribution, we argue that $b(X, C_t) \overset{a.s.}{\to} \theta^e_x$. Suppose this was not true. This implies that the probability that $P(\lim_{t\to\infty} b(X, C_t) \neq \theta^e_x | X = x, E = e) = \beta > 0$. Since $X = x$, $E = e$ occurs with a finite probability (as X and E are discrete-valued and x, e is in the support) say α, then $\alpha\beta$ fraction of sequences of $b(X, C_t)$ do not converge to θ^e_x, which contradicts the assumption that $b(X, C_t) \overset{a.s.}{\to} \theta^E_X$.

Consider a (x, θ) from the support of (X, θ^E_X), where X is the current query and E is the environment from which X and context preceeding it is sampled. Let us consider the distribution $P(Y|X, b(X, C_t))$.

$$P(Y = y|X = x, b(X, C_t) = \theta) = \frac{P(Y = y, X = x, b(X, C_t) = \theta)}{P(X = x, b(X, C_t) = \theta)} \tag{11}$$

We simplify $\lim_{t\to\infty} P(Y|X, b(X, C_t))$ below.

$$\lim_{t\to\infty} P(Y = y|X = x, b(X, C_t) = \theta) = \frac{\lim_{t\to\infty} P(Y = y, X = x, b(X, C_t) = \theta)}{\lim_{t\to\infty} P(X = x, b(X, C_t) = \theta)} \tag{12}$$

We show that the limits of the numerator and denominator exist (and non-zero for the denominator) and we simplify these separately below.

$$\begin{aligned}
&\lim_{t\to\infty} P(Y=y, X=x, b(X,C_t)=\theta) = \lim_{t\to\infty} \sum_e P(Y=y, X=x, E=e, b(X,C_t)=\theta) \\
&= \sum_e P(Y=y|X=x, E=e) \lim_{t\to\infty} P(X=x, E=e, b(X,C_t)=\theta) \\
&= \sum_e P(Y=y|X=x, E=e) P(X=x, E=e) \lim_{t\to\infty} P(b(X,C_t)=\theta|X=x, E=e)
\end{aligned} \tag{13}$$

In the simplification above, we firstly used the fact that we can interchange sum and limits, this is true because e only takes finitely many values. In the simplification above, we also use the fact $Y \perp C_t | X, E$. Since $b(X, C_t)$ converges to θ_x^e almost surely, the distribution $\lim_{t\to\infty} P(b(X,C_t) = \theta|X=x, E=e)$ takes a value one if $\theta = \theta_x^e$ and zero otherwise. As a result, the above expression becomes

$$\lim_{t\to\infty} P(Y=y, X=x, b(X,C_t)=\theta) = \sum_{e\in\mathcal{E}_{x,\theta}} P(Y=y|X=x, E=e) P(X=x, E=e). \tag{14}$$

where $\mathcal{E}_{x,\theta}$ is the set of all the environments observed conditional on $X=x$ with $\theta_x^e = \theta$. Observe that all the environments in $\mathcal{E}_{x,\theta}$ have the same $P(Y=1|X=x, E=e)$ given by $h^\star(x,\theta)$. We can write

$$\lim_{t\to\infty} P(Y=1, X=x, b(X,C_t)=\theta) = h^\star(x,\theta) \sum_{e\in\mathcal{E}_{x,\theta}} P(X=x, E=e). \tag{15}$$

We simplify $\lim_{t\to\infty} P(X=x, b(X,C_t)=\theta)$ in a similar manner to obtain

$$\lim_{t\to\infty} P(X=x, b(X,C_t)=\theta) = \sum_{e\in\mathcal{E}_{x,\theta}} P(X=x, E=e). \tag{16}$$

Observe that the denominator is positive and not zero because x, θ is in support of X, θ_X^E. We use equation 15 and equation 16 to obtain

$$\begin{aligned}
\lim_{t\to\infty} P(Y=1|X=x, b(X,C_t)=\theta) &= \frac{\lim_{t\to\infty} P(Y=1, X=x, b(X,C_t)=\theta)}{\lim_{t\to\infty} P(X=x, b(X,C_t)=\theta)} \\
&= \frac{h^\star(x,\theta)\sum_{e\in\mathcal{E}_{x,\theta}} P(X=x, E=e)}{\sum_{e\in\mathcal{E}_{x,\theta}} P(X=x, E=e)} = h^\star(x,\theta).
\end{aligned} \tag{17}$$

Therefore,

$$\lim_{t\to\infty} P(Y=1|X=x, b(X,C_t)=\theta) = P(Y=1|X=x, E=e). \tag{18}$$

where e is any environment in $\mathcal{E}_{x,\theta}$, i.e., it is in the support of data sampled with $X=x$ and that also satisfies $\theta_x^e = \theta$.

$$\begin{aligned}
&\lim_{t\to\infty} H(Y|X, b(X,C_t)) = \sum_{x,\theta} \lim_{t\to\infty} P(X=x, b(X,C_t)=\theta) \lim_{t\to\infty} H(Y|X=x, b(X,C_t)=\theta) \\
&\sum_{x,\theta} \Big(\sum_{\tilde{e}\in\mathcal{E}_{x,\theta}} P(X=x, E=\tilde{e}) \Big) \lim_{t\to\infty} H(Y|X=x, b(X,C_t)=\theta)
\end{aligned} \tag{19}$$

In the above simplification, we again swap limits and sum because the summation is over a finite set of values. From equation 18, it follows that $\lim_{t\to\infty} H(Y|X=x, b(X,C_t)=\theta) = H(Y|X=$

$x, E = e)$, where e is any environment in $\mathcal{E}_{x,\theta}$. We use this in the above to get

$$\begin{aligned}\lim_{t\to\infty} H(Y|X, b(X, C_t)) &= \sum_{x,\theta} \Big(\sum_{\tilde{e}\in\mathcal{E}_{x,\theta}} P(X = x, E = \tilde{e})\Big) H(Y|X = x, E = e) \\ &= \sum_{x,\theta} \Big(\sum_{\tilde{e}\in\mathcal{E}_{x,\theta}} P(X = x, E = \tilde{e})\Big) H(Y|X = x, E = \tilde{e}) \\ &= \sum_{x,\tilde{e}} P(X = x, E = \tilde{e}) H(Y|X = x, E = \tilde{e}) = H(Y|X, E).\end{aligned} \tag{20}$$

We combine the above with equation 10 to obtain $\lim_{t\to\infty} H(Y|X, C_t) = H(Y|X, E)$. Finally, observe that if $I(Y; E|X) > 0 \implies H(Y|X, E) < H(Y|X)$. Since $\lim_{t\to\infty} H(Y|X, C_t) = H(Y|X, E)$, ICRM impoves over ERM that attains a cross-entropy loss of $H(Y|X)$.

This completes the proof.

□

We now extend the argument to setting beyond discrete random variables. In particular, we consider settings where $X, E, b(X, C_t)$ can be either discrete or continuous random variables. In the notation to follow, we use dP to denote the Radon-Nikodym derivatives. For discrete random variable, the Radon-Nikodym derivatives correspond to the standard probability mass function and for continuous random variables it would correspond to standard probability density functions. We operate under some regularity assumptions. We assume that the support of E has a finite volume and the support of $(X, b(X, C_t))$ has a finite volume for all t. Further, we assume that the Radon-Nikodym derivative of the joint $dP(X = x, E = e, b(X, C_t) = \theta)$ is bounded above. While much of the proof that follows is same as the previous proof, we repeat the arguments for completeness.

Theorem 4. *Let $h^\star(x, \theta_x^e)$ describe $dP(Y = 1 \mid X = x, E = e)$ for all $e \in \mathcal{E}$. Further, we assume the existence of an* amortization function $b(X, C_t) \overset{a.s.}{\to} \theta_X^E$. *Then,* ICRM *zooms-in on the environment risk minimizer and achieves a cross-entropy loss over the training distribution*

$$\lim_{t\to\infty} H(Y \mid X, C_t) = H(Y \mid X, E).$$

Further, if $I(Y; E \mid X) > 0$, ICRM *has better performance than the global risk minimizer.*

Proof. Since each (X_j, Y_j) is sampled independently given a training environment E, we can conclude $I(Y; C_t|X, E) = 0$. Therefore,

$$I(Y; C_t|X, E) = 0 \implies H(Y|X, E) = H(Y|X, E, C_t).$$

Observe that for all $t \in \mathbb{Z}_+$

$$H(Y|X, E) = H(Y|X, E, C_t) \le H(Y|X, C_t) \le H(Y|X, b(X, C_t)), \tag{21}$$

where $\mathbb{Z}_+$ is the set of all positive integers. The first inequality in the above follows from the fact that conditioning reduces entropy. For the second inequality, we use the following property. Consider U, V as two random variables and define $W = a(V)$. Observe that $I(U; W|V) = 0 \implies H(U|V) = H(U|V, W) \le H(U|W)$.

Since the inequality above equation 21 holds for all t, we obtain

$$H(Y|X, E) \le \lim_{t\to\infty} H(Y|X, C_t) \le \lim_{t\to\infty} H(Y|X, b(X, C_t)). \tag{22}$$

In the above, we use the following property. If $a_n \le b_n, \forall n \in \mathbb{Z}_+$ and $\lim_{n\to\infty} a_n$ and $\lim_{n\to\infty} b_n$ exist, then $\lim_{n\to\infty} a_n \le \lim_{n\to\infty} b_n$. In what follows, we will show that both the limits $\lim_{t\to\infty} H(Y|X, C_t)$ and $\lim_{t\to\infty} H(Y|X, b(X, C_t))$ exist. First observe that $H(Y|X, C_{t+1}) \le H(Y|X, C_t)$ for all t as a result the sequence is decreasing bounded below by 0 and thus from Monotone convergence theorem $\lim_{t\to\infty} H(Y|X, C_t)$ exists. Next, we will show that $\lim_{t\to\infty} H(Y|X, b(X, C_t)) = H(Y|X, E)$. We will then combine it with equation 22 to obtain what we intend to prove, i.e., $\lim_{t\to\infty} H(Y|X, C_t) = H(Y|X, E)$.

For each $X = x$ and $E = e$ in the support except over a set of probability measure zero, we argue that $b(X, C_t) \overset{a.s.}{\to} \theta_x^e$. Suppose this was not true. Define Γ to be the set of values of x, e for which $b(X, C_t) \overset{a.s.}{\not\to} \theta_x^e$. Let $P((X, E) \in \Gamma) > 0$ and the probability that $P(\lim_{t\to\infty} b(X, C_t) \neq \theta_X^E | (X, E) \in \Gamma) > 0$. If this is true then $P(\lim_{t\to\infty} b(X, C_t) \neq \theta_x^e) > 0$ contradicts the fact that $b(X, C_t) \overset{a.s.}{\to} \theta_X^E$. Therefore, $P((X, E) \in \Gamma) = 0$.

Consider a (x, θ) from the support of (X, θ_X^E) except from Γ, where X is the current query and E is the environment from which X and context preceeding it is sampled. Let us consider the distribution $dP(Y|X, b(X, C_t))$.

$$dP(Y = y|X = x, b(X, C_t) = \theta) = \frac{dP(Y = y, X = x, b(X, C_t) = \theta)}{dP(X = x, b(X, C_t) = \theta)} \tag{23}$$

We simplify $\lim_{t\to\infty} dP(Y = y|X = x, b(X, C_t) = \theta)$ below.

$$\lim_{t\to\infty} dP(Y = y|X = x, b(X, C_t) = \theta) = \frac{\lim_{t\to\infty} dP(Y = y, X = x, b(X, C_t) = \theta)}{\lim_{t\to\infty} dP(X = x, b(X, C_t) = \theta)} \tag{24}$$

We simplify the numerator and the denominator of the above separately.

$$\begin{aligned}
&\lim_{t\to\infty} dP(Y = y, X = x, b(X, C_t) = \theta) = \lim_{t\to\infty} \int_e dP(Y = y, X = x, E = e, b(X, C_t) = \theta) \\
&= \int_e dP(Y = y|X = x, E = e) \lim_{t\to\infty} dP(X = x, E = e, b(X, C_t) = \theta) \\
&= \int_e dP(Y = y|X = x, E = e) dP(X = x, E = e) \lim_{t\to\infty} dP(b(X, C_t) = \theta|X = x, E = e)
\end{aligned} \tag{25}$$

In the above, we use dominated convergence theorem [4] to swap limit and the integrals (to use dominated convergence theorem, we use the fact that the $dP(X = x, E = e, b(X, C_t) = \theta)$ is bounded and support of E has a finite volume). In the simplification above, we also use the fact $Y \perp C_t|X, E$. Since $b(X, C_t)$ converges to θ_x^e almost surely, the distribution $\lim_{t\to\infty} dP(b(X, C_t) = \theta|X = x, E = e)$ evaluates to probability one when $\theta = \theta_x^e$ and is zero otherwise. As a result, the above expressions become

$$\lim_{t\to\infty} dP(Y = y, X = x, b(X, C_t) = \theta) = \int_{e\in\mathcal{E}_{x,\theta}} dP(Y = y|X = x, E = e) dP(X = x, E = e). \tag{26}$$

where $\mathcal{E}_{x,\theta}$ is the set of all the environments observed conditional on $X = x$ with $\theta_x^e = \theta$. Observe that all the environments in $\mathcal{E}_{x,\theta}$ have the same $dP(Y = 1|X = x, E = e)$ given by $h^\star(x, \theta)$. Similarly,

$$\lim_{t\to\infty} dP(X = x, b(X, C_t) = \theta) = \int_{e\in\mathcal{E}_{x,\theta}} dP(X = x, E = e). \tag{27}$$

As a result, we can write

$$\lim_{t\to\infty} dP(Y = 1, X = x, b(X, C_t) = \theta) = h^\star(x, \theta) \int_{e\in\mathcal{E}_{x,\theta}} dP(X = x, E = e).$$

We use this to obtain

$$\lim_{t\to\infty} dP(Y=1|X=x, b(X,C_t)=\theta) = \frac{\lim_{t\to\infty} dP(Y=1, X=x, b(X,C_t)=\theta)}{\lim_{t\to\infty} dP(X=x, b(X,C_t)=\theta)} \\ = \frac{h^\star(x,\theta)\int_{e\in\mathcal{E}_{x,\theta}} dP(X=x,E=e)}{\int_{e\in\mathcal{E}_{x,\theta}} dP(X=x,E=e)} = h^\star(x,\theta). \tag{28}$$

Therefore,

$$\lim_{t\to\infty} dP(Y=y|X=x, b(X,C_t)=\theta) = dP(Y=y|X=x,E=e). \tag{29}$$

where e is any environment that is in the support of data sampled with $X=x$ and that also satisfies $\theta_x^e=\theta$.

$$\lim_{t\to\infty} H(Y|X,b(X,C_t)) = \int_{x,\theta} \lim_{t\to\infty} dP(X=x,b(X,C_t)=\theta) \lim_{t\to\infty} H(Y|X=x,b(X,C_t)=\theta) \\ = \int_{x,\theta} \Big(\int_{\tilde{e}\in\mathcal{E}_{x,\theta}} dP(X=x,E=\tilde{e})\Big) \lim_{t\to\infty} H(Y|X=x,b(X,C_t)=\theta) \tag{30}$$

In the above, we use dominated convergence theorem to swap the limits and integrals (Recall that $dP(X=x,E=e,b(X,C_t)=\theta)$ is bounded say by say ς and the volume of the support of E is bounded say by φ. As a result, $dP(X=x,b(X,C_t)=\theta)H(Y|X=x,b(X,C_t)=\theta) \le \varsigma\varphi\log(2)$.). From equation 29, it follows that $\lim_{t\to\infty} H(Y|X=x,b(X,C_t)=\theta) = H(Y|X=x,E=e)$, where e is any environment in $\mathcal{E}_{x,\theta}$. We use this in the above to get

$$\lim_{t\to\infty} H(Y|X,b(X,C_t)) = \int_{x,\theta} \Big(\int_{\tilde{e}\in\mathcal{E}_{x,\theta}} dP(X=x,E=\tilde{e})\Big) H(Y|X=x,E=e) \\ = \int_{x,\theta} \Big(\int_{\tilde{e}\in\mathcal{E}_{x,\theta}} dP(X=x,E=\tilde{e})\Big) H(Y|X=x,E=\tilde{e}) \\ = \int_{x,\tilde{e}} dP(X=x,E=\tilde{e}) H(Y|X=x,E=\tilde{e}) = H(Y|X,E). \tag{31}$$

We combine the above with equation 22 to obtain $\lim_{t\to\infty} H(Y|X,C_t) = H(Y|X,E)$. Finally, observe that if $I(Y;E|X) > 0 \implies H(Y|X,E) < H(Y|X)$. Since $\lim_{t\to\infty} H(Y|X,C_t) = H(Y|X,E)$, ICRM impoves over ERM that attains a cross-entropy loss of $H(Y|X)$.

This completes the proof.

□

A.3 Proof of Theorem 2

Theorem 2 (Partial iid zoom-in)**.** *Suppose the joint distribution $((X_1,\cdots X_t),(Y_1,\ldots,Y_t),E)$ is Markov with respect to a Bayesian network. The query X and the environment E are statistically dependent and form the Markov blanket of Y. Then* ICRM *partially zooms-in on the environment risk minimizer, improving over the performance of the global empirical risk minimizer in terms of the cross-entropy loss. Further, the improvement is strictly monotonic in context length t.*

Proof. Let us consider the setting where the context is of length one. We denote the current query as X with corresponding label Y and environment E. The example in the context is $\tilde{X}$ which has corresponding label $\tilde{Y}$ and it shares the same environment E. Recall that as part of the context, the learner only sees $\tilde{X}$ and not $\tilde{Y}$. Both Y and E are real-valued scalars and X is a d dimensional vector.

Following the assumption in the theorem, the distribution of $(\tilde{X},\tilde{Y},X,Y,E)$ is Markov with respect to a Bayesian network. We first establish that E cannot be a child of any variable in the directed

acyclic graph (DAG). The assumption $(X, Y) \perp (\tilde{X}, \tilde{Y})|E$ implies $X \perp \tilde{X}|E$ and $Y \perp \tilde{Y}|E$. Suppose E is a child variable of Y. Due to the symmetry, (X, Y, E) and $(\tilde{X}, \tilde{Y}, E)$ follow the same distribution. As a result, E is also a child variable of $\tilde{Y}$, which implies $Y \not\perp \tilde{Y}|E$ (since E is a collider on the path from Y to $\tilde{Y}$). This contradicts $Y \perp \tilde{Y}|E$. Suppose E is a child variable of some component of X say X^i. Due to symmetry, E is also a child variable of $\tilde{X}^i$, which implies $X^i \not\perp \tilde{X}^i|E$. This contradicts $X \perp \tilde{X}|E$. Therefore, E cannot be a child of any of the variables in the DAG.

Since both X and E form the Markov blanket of Y, there are two possible cases. Either E is directly connected to Y or E is connected to Y through some element of X.

In the first case, E can only have an arrow into Y and not the other way around as E is not a child of any other node. Let us consider the setting when E is one of the parents of Y and denote it as $E \to Y$. Since X ($\tilde{X}$) is on the Markov Blanket of Y ($\tilde{Y}$), we claim that each component of X is either a parent of Y or a child of Y. Suppose this was not the case. This implies that there exists a component of X say X^i, which is on the Markov Blanket as a parent of E. But that would make E a child of Y. However, E cannot be a child variable as shown above. As a result, each component of X is either a parent or a child of Y. We now consider two subcases.

Let us consider the setting when there exists a child X^i of Y. Observe that $\tilde{X}^i$ is a child of $\tilde{Y}$ and it has a path to E and as a result it has a path to Y. This path from elements of $\tilde{X}^i$ to $\tilde{Y}$ passes through E. This path has no colliders and does not contain any element of X on it (We show this case in Figure 2a). As a result, $Y \not\perp \tilde{X}^i|X$. Thus $I(Y; \tilde{X}|X) > 0$ (use chain rule of mutual information).

Let us consider the other setting when each X^i is a parent of Y (shown in Figure 2b). In this case, E has to have a path to some element of X, say X^j as otherwise $E \perp X$, which contradicts the assumption that $E \not\perp X$. Consider the path $\tilde{X}^j$ to E to Y. Observe that this path is not blocked. As a result, $I(Y; \tilde{X}|X) > 0$.

Let us consider the other possibility when Y is connected to E through X. Here the only way this is possible is if some element of X say X^i is a child of Y and E is a parent of that element (as shown in Figure 2c). Therefore, we know that $\tilde{X}^i$ is connected to Y through E and X^i.

Observe that this path from $\tilde{X}^i$ to Y is not blocked as X^i is a collider. Therefore, $I(Y; \tilde{X}|X) > 0$. We showed the result so far assuming that the context length was one. Suppose that the context has $k-1$ examples denoted as $C_k = [X_1 \cdots, X_{k-1}]$. The chain rule of mutual information tells us $I(Y; C_k|X) = I(Y; X_{k-1}|X) + I(Y; C_{k-1}|X, X_{k-1})$. The proof above already demonstrates that the first term $I(Y; X_{k-1}|X)$ is strictly positive. Since mutual information is non-negative, we can conclude that $I(Y; C_k|X) > 0$.

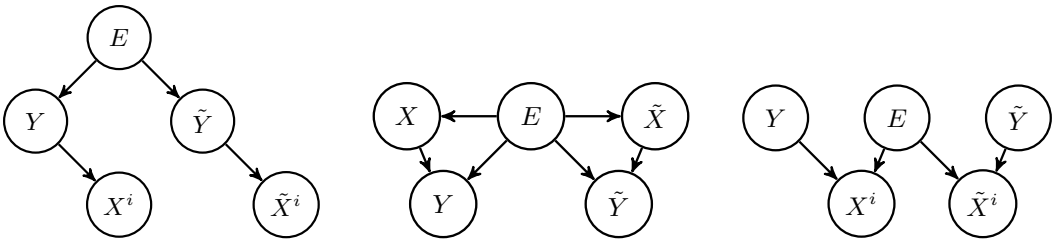
Next, we want to argue that entropy strictly reduces as context length increases. In other words,

$$H(Y|X, C_k) < H(Y|X, C_{k-1}) \iff I(Y; X_k|X, C_{k-1}) > 0.$$

We want to show $Y \not\perp X_k|(X, C_{k-1})$. In the proof above, we had three cases shown in Figure 2. In each of these cases, we argued that the path from X_k to Y is not blocked. Even if we condition on contexts C_{k-1} this continues to be the case. In the first two cases, the path from X_k to Y is direct and does not contain any element from the conditioning set. In the third case, the direct path involves a collider X from the conditioning set and thus is also not blocked. As a result, $Y \not\perp X_k|(X, C_{k-1})$. This completes the proof.

□

Remark on the Theorem 2 It is possible to extend Theorem 2 to the case when only a subset of X and E form the Markov blanket. Observe that the analysis of Case a) and Case c) in Figure 2a, Figure 2c does not change. The analysis of Case b) is more nuanced now. In Case b), we used the fact that E is connected to X that is on the Markov blanket. This need not be the case if only a subset of X is on the Markov blanket. Suppose X_{MB} denote the set of X that are on the Markov Blanket. If E is connected to any member of X_{MB}, the same analysis as Case b) continues to hold. Consider the case when E is connected to some other member of X that is not in X_{MB}. Denote this member as X^i. Observe that the same element $\tilde{X}^i$ from $\tilde{X}$ will have a direct path into Y through E that is not blocked. As a result, even in this case conditioning on $\tilde{X}$ helps.



(a) Case 1. (b) Case 2. (c) Case 3.

Figure 2: Illustrating the different key cases for Theorem 2.

A.4 Proof of Theorem 3

Theorem 3 (Full ood zoom-in) *Consider data triplets* (x, y, e) generated from $z \sim \mathcal{N}(\mu_e^y, \Sigma_e^y)$ *and* $x \leftarrow g(z)$, *for all environments* $e \in \mathcal{E}$, *where* g *is the identity map. There exists an ICL algorithm that in the limit of infinitely long contexts produces Bayes optimal predictions for all the test environments that fall in the Voronoi cells of the training environments.*

Proof. The learning algorithm works as follows. For each e, y pair in the training data, define the set of $x's$ as $\mathcal{D}_x^{e,y}$. Maximize the likelihood of $\mathcal{D}_x^{e,y}$ assuming that the underlying distribution is Gaussian. This can be stated as

$$\hat{\mu}_e^y, \hat{\Sigma}_e^y = \arg\min_{\mu_e^y, \Sigma_e^y} \Big(\sum_{x \in \mathcal{D}_x^{e,y}} \Big[\|x - \mu_e^y\|^2_{(\Sigma_e^y)^{-1}} \Big] - \log(\mathsf{det}(\Sigma_e^y)) \Big).$$

The solution to the above are standard sample mean based estimators of means and covariance. Also, use a sample mean based estimator to estimate the probability of each class in environment e and denote it as $\hat{p}_e^y$. Define $\hat{\gamma}_e = [(\hat{p}_e^0, \hat{\mu}_e^0, \hat{\Sigma}_e^0), (\hat{p}_e^1, \hat{\mu}_e^1, \hat{\Sigma}_e^1)]$. The model at test time works as follows.

- We are given samples $\mathcal{D}_x^{e'}$ at test time from some environment $e' \in \mathcal{E}_{te}$. Estimate the parameters of Gaussian mixture model with two mixture components to maximize the likelihood of observing $\mathcal{D}_x^{e'}$. We denote the estimated parameters as $\theta_{e'} = [p_{e'}, \mu_{e'}, \Sigma_{e'}, \tilde{p}_{e'}, \tilde{\mu}_{e'}, \tilde{\Sigma}_{e'}]$. Define a permutation of $\theta_{e'}$ as $\beta_{e'} = [\tilde{p}_{e'}, \tilde{\mu}_{e'}, \tilde{\Sigma}_{e'}, p_{e'}, \mu_{e'}, \Sigma_{e'}]$.
- Find the closest environment to the estimated parameters in the training set.

$$\min_{e \in \mathcal{E}_{tr}} \Big(\min\{\|\theta_{e'} - \hat{\gamma}_e\|, \|\beta_{e'} - \hat{\gamma}_e\|\} \Big) \tag{32}$$

  Suppose $\tilde{e}$ is the closest training environment that solves the above. If $\theta_{e'}$ is closer to $\hat{\gamma}_{\tilde{e}}$ than $\beta_{e'}$, then $p_{e'}, \mu_{e'}, \Sigma_{e'}$ correspond to the label 0 and $\tilde{p}_{e'}, \tilde{\mu}_{e'}, \tilde{\Sigma}_{e'}$ correspond to the label 1. For the query x, the probability assigned to label 0 is

$$c(x) = \frac{p_{e'} e^{-\|x - \mu_{e'}\|^2_{(\Sigma_{e'})^{-1}}}}{p_{e'} e^{-\|x - \mu_{e'}\|^2_{(\Sigma_{e'})^{-1}}} + \tilde{p}_{e'} e^{-\|x - \tilde{\mu}_{e'}\|^2_{(\tilde{\Sigma}_{e'})^{-1}}}}.$$

  If $\beta_{e'}$ is closest to this environment, then $p_{e'}, \mu_{e'}, \Sigma_{e'}$ correspond to the label 1 and $\tilde{p}_{e'}, \tilde{\mu}_{e'}, \tilde{\Sigma}_{e'}$ is the label 0. For the query x, the probability assigned to label 0 is $1 - c(x)$.

For the training environments, in the limit of infinitely long contexts the estimated parameters take exact values, i.e., $\hat{\gamma}_e = \gamma_e$, for all $e \in \mathcal{E}_{tr}$.

For the test environment, the true set of parameters that generate the data are $\gamma_{e'}$, where $\gamma_{e'} = \big[(p_{e'}^0, \mu_{e'}^0, \Sigma_{e'}^0), (p_{e'}^1, \mu_{e'}^1, \Sigma_{e'}^1)\big]$. Define the permutation of $\gamma_{e'}$ as $\delta_{e'} = \big[(p_{e'}^1, \mu_{e'}^1, \Sigma_{e'}^1), (p_{e'}^0, \mu_{e'}^0, \Sigma_{e'}^0)\big]$.

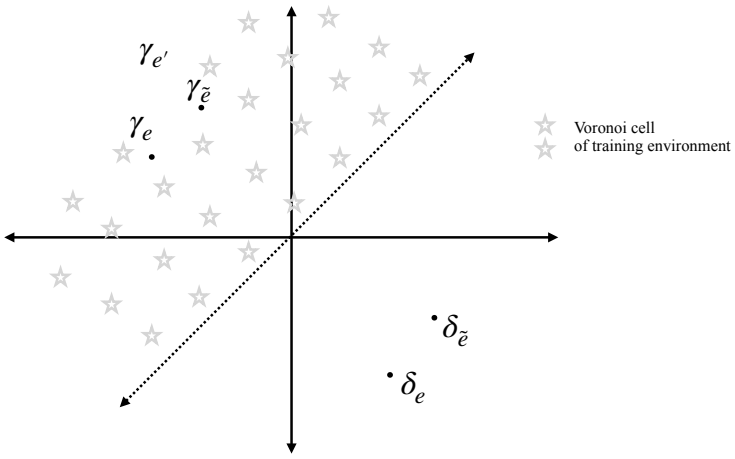


Figure 3: Illustration of Voronoi cells of training environment.

There can be two types of test environments. One in which the mean and covariance for both classes are identical. The method above assigns a probability of $\frac{1}{2}$ to both the classes, which is the Bayes optimal prediction. Let us consider the latter environments, where the class conditional parameters for x are not the same. In the limit of infinitely long contexts at test time, there are two possible values $\theta_{e'}$ can take, either $\theta_{e'} = \gamma_{e'}$ or $\theta_{e'} = \delta_{e'}$. This follows from identifiability of Gaussian mixtures, [55].

Consider the first case, $\theta_{e'} = \gamma_{e'}$. In this case, the equation 32 becomes

$$\min_{e \in \mathcal{E}_{tr}} \Big(\min\{\|\gamma_{e'} - \gamma_e\|, \|\delta_{e'} - \gamma_e\|\} \Big).$$

Suppose some environment $\tilde{e}$ solves the above optimization. Following the assumption in we know that $\gamma_{e'}$ falls in the Voronoi region of some $\gamma_{\tilde{e}}$ and thus $\gamma_{e'}$ is closer to $\gamma_{\tilde{e}}$ than $\delta_{\tilde{e}}$ (see Figure 3). As a result, $p^0_{e'}, \mu^0_{e'}, \Sigma^0_{e'}$ is associated with class 0, which is actually correct and thus the final predictor would match the Bayes optimal predictor for the test environment. In the second case, $\theta_{e'} = \delta_{e'}$. Therefore, $\beta_{e'} = \gamma_{e'}$ and $p^1_{e'}, \mu^1_{e'}, \Sigma^1_{e'}$ would be correctly associated with class one thus leading to Bayes optimal predictions. This completes the argument we set out to prove.

We now briefly explain how the method fails if test parameter is outside the Voronoi cell of training parameters. Suppose $\theta_{e'} = \gamma_{e'}$ but $\gamma_{e'}$ is in Voronoi region of some δ_e. In this case, $\beta_{e'}$ would be closest to γ_e and $p^0_{e'}, \mu^0_{e'}, \Sigma^0_{e'}$ would be incorrectly associated with class 1. This shows that beyond the Voronoi region the proposed algorithm fails.

□

A.5 Extension of Theorem 3

In the previous theorem, we assumed that g is identity. We now describe how the result can be extended to general non-linear mixing maps g. For this result, we leverage the theoretical results from identifiable variational autoencoders (i-VAE) [20].

A short review of identifiable variational autoencoders We are provided with observations x's that are generated from a latent variable z using an injective map g, where $x \leftarrow g(z)$. The theory of i-VAE provides with a method and the conditions under which the underlying true latent variables z can be identified up to permutation and scaling. In i-VAEs, it is assumed that along with each sample x, we are provided with auxiliary information, which they term as u. For our results, auxiliary information is available to us in the form of the environment index and the label of the data point.

In the theory of i-VAE, the distribution of the latent variables are assumed to follow a conditionally factorial exponential distribution stated as follows.

$$p_{T,\lambda}(z|u) = \prod_i \frac{Q_i(z_i)}{M_i(u)} \exp\left[\sum_{j=1}^{k} T_{i,j}(z_i)\lambda_{i,j}(u)\right] \tag{33}$$

where $T_i = (T_{i,1}, \cdots, T_{i,k})$ are the sufficient statistics, $\lambda_i(u) = (\lambda_{i,1}(u), \cdots, \lambda_{i,k}(u))$ are the parameters of the distribution that vary with u, Q_i is a base measure and M_i is a normalizing constant. We concatenate T_i's and $\lambda_i's$ across d latent dimensions to make construct dk dimensional vectors denoted as $\lambda(u)$ and $T(z)$. Thus the data generation process is summarized as

$$\begin{aligned} z &\sim p_{T,\lambda}(\cdot|u), \\ x &\leftarrow g(z), \end{aligned} \tag{34}$$

where g, T, λ are the parameters. We now revisit the data generation process that we consider and explain how it falls under the umbrella of the data generation processes considered in i-VAE. For all $e \in \mathcal{E}$,

$$\begin{aligned} z|y, e &\sim \mathcal{N}(\mu_e^y, \Sigma_e^y), \\ x &\leftarrow g(z), \end{aligned} \tag{35}$$

where the latent variables z are sampled conditional on the label y and environment e from a Normal distribution whose mean and covariance depend on both y, e. Define $\mathcal{X}$ as the image of g, i.e., $\mathcal{X} = g(\mathbb{R}^d)$. We further assume that the covariance matrix has a diagonal structure as stated below.

Assumption 1. *Each Σ_e^y is a diagonal matrix.*

Since Σ_e^y is a diagonal matrix, we denote the i^{th} diagonal element as $(\sigma_e^y(i))^2$. Similarly, the i^{th} component of μ_e^y is denoted as $\mu_e^y(i)$. Observe that the distribution of z conditional on y, e belongs to the family conditionally factorial exponential distributions studied in i-VAE. If we substitute $Q_i(z_i) = \frac{1}{\sqrt{2\pi}}$, $M_i(y,e) = e^{\left((\mu_e^y(i))^2/(\sigma_e^y(i))^2\right)}$, $\lambda_{i,1}(y,e) = \frac{2\mu_e^y(i)}{(\sigma_e^y(i))^2}$, $\lambda_{i,2}(y,e) = -\frac{1}{(\sigma_e^y(i))^2}$, $T_{i,1}(z) = z$ and $T_{i,2}(z) = z^2$, then we obtain the distribution of z described by equation 35.

Definition 1. *We define an equivalence relation between sets of parameters of the model as follows.*

$$(g, T, \lambda) \sim (\tilde{g}, \tilde{T}, \tilde{\lambda}) \iff \exists A, c \mid T(g^{-1}(x)) = A\tilde{T}(\tilde{g}^{-1}(x)) + c, \forall x \in \mathcal{X}. \tag{36}$$

If A is invertible, then we denote the relation by $\sim_A$. If A is a block permutation matrix, then we denote it by $\sim_P$.

Theorem 5. *Assume that the data is sampled from the data generation in equation 34 according to with parameters (g, T, λ). Assume the following holds*

- *The mixing function g is injective*
- *The sufficient statistics $T_{i,j}$ are differentiable almost everywhere, and $(T_{i,j})_{1 \le j \le k}$ are linearly independent on any subset of $\mathcal{X}$ of measure greater than zero.*
- *There exists $dk + 1$ distinct points $u_0, \cdots, u_{dk}$ such that the matrix*

  $$L = (\lambda(u_1) - \lambda(u_0), \cdots, \lambda(u_{dk}) - \lambda(u_0))$$

  of size $dk \times dk$ is invertible.

then the parameters (g, T, λ) are $\sim_A$ identifiable.

Theorem 6. *Assume the hypotheses of the Theorem 5 holds, and $k \ge 2$. Further assume:*

- *The sufficient statistics $T_{i,j}$ are twice differentiable.*
- *The mixing function g has all second order cross derivatives.*

then the parameters (g, T, λ) are $\sim_P$ identifiable.

We can leverage the above two theorems (Theorem 5, Theorem 6 and Theorem 4 from [26]) and arrive at the following corollary for the Gaussian data generation process from equation 35.

Theorem 7. *If the data generation process follows equation 35, where g is injective and has all second order cross derivatives. Suppose there exist $2d+1$ points $u^0 = (y_0, e_0), \cdots, u^{2d} = (y_{2d}, e_{2d})$ in the support of (y, e) observed in training distribution such that*

$$(\lambda(u_1) - \lambda(u_0), \cdots, \lambda(u_{2d}) - \lambda(u_0))$$

is invertible. If $p_{g,T,\lambda}(\cdot|y, e) = p_{\tilde{g},\tilde{T},\tilde{\lambda}}(\cdot|y, e)$ for all y, e in the support of (y, e) in the training distribution, then $\tilde{z} = \Lambda\Pi z + r$, where $\tilde{z} = \tilde{g}^{-1}(x)$ and $z = g^{-1}(x)$.

Proof. We equate the probability of observations x under two models g, T, λ and $\tilde{g}, \tilde{T}, \tilde{\lambda}$ for each y, e. Consider a $z \sim p_{T,\lambda}(\cdot|y, e)$ and the corresponding $x = g(z)$. These x's follow $p_{\tilde{g},\tilde{T},\tilde{\lambda}}(\cdot|y, e)$ since $p_{g,T,\lambda}(\cdot|y, e) = p_{\tilde{g},\tilde{T},\tilde{\lambda}}(\cdot|y, e)$. Define $\tilde{z} = \tilde{g}^{-1}(x)$ and these $\tilde{z}$ follow $p_{\tilde{T},\tilde{\lambda}}(\cdot|y, e)$. We can write $\tilde{z} = a(z)$, where $a = \tilde{g}^{-1} \circ g$.

Observe $p_z(z|y, e) = p_{\tilde{z}}(a(z)|y, e)\mathsf{det}(Da(z))$ and

$$\begin{aligned}
\log p_z\big(z|y_k, e_k\big) &= \log\big(p_{\tilde{z}}(a(z)|y_k, e_k)\big) + \log \mathsf{det}(Da(z)), \\
\log p_z\big(z|y_0, e_0\big) &= \log\big(p_{\tilde{z}}(a(z)|y_0, e_0)\big) + \log \mathsf{det}(Da(z)), \qquad (37) \\
\log p_z\big(z|y_k, e_k\big) - \log\big(p_z(z|y_0, e_0)\big) &= \log\big(p_{\tilde{z}}(a(z)|y_k, e_k)\big) - \log\big(p_{\tilde{z}}(a(z)|y_0, e_0)\big).
\end{aligned}$$

Substituting the exponential form we obtain that

$$T(z)^\top[\lambda(y_k, e_k) - \lambda(y_0, e_0))] = T(\tilde{z})^\top[\tilde{\lambda}(y_k, e_k) - \tilde{\lambda}(y_0, e_0))]$$

If we use sufficient variability conditions, we obtain $T(z) = AT(\tilde{z}) + c$. We now use the fact that sufficient statistics $T(z) = (z, z^2)$ are minimal to conclude that

$$T(z) = AT(\tilde{z}) + c$$

where A is invertible. In the above, we use the line of reasoning used in in the proof of Theorem 4 in [26].

After this point, we leverage Theorem 6 to conclude that

$$T_i(z_i) = AT_j(\tilde{z}_j) + c.$$

We can expand the above to write

$$\begin{bmatrix} \tilde{z}_j \\ \tilde{z}_j^2 \end{bmatrix} = D \begin{bmatrix} z_i \\ z_i^2 \end{bmatrix} + e.$$

Note that the above relationship holds for all $z \in \mathcal{Z}$. If $\tilde{z}_j$ depends on z_i^2, then $\tilde{z}_j^2$ would be a degree four polynomial in z_i and it would be equated to a degree 2 polynomial z_i stated in the RHS. This cannot be true for all z_i in the support. As a result, $\tilde{z}_j$ is a scalar multiple of z_i. Since for every i there is such a j, it follows that $\tilde{z} = \Lambda\Pi z + r$.

□

Theorem 8. ***(Zoom-in [ood])*** *Consider the data generation process in equation 35. We make a few additional assumptions on the data generation stated below.*

- *Each Σ_e^y is a diagonal matrix*
- *There exist $2d + 1$ points $u^0 = (y_0, e_0), \cdots, u^{2d} = (y_{2d}, e_{2d})$ in the support of (y, e) observed in training distribution such that*

  $$(\lambda(u_1) - \lambda(u_0), \cdots, \lambda(u_{2d}) - \lambda(u_0))$$

  is invertible.

- *g is injective and has all second order cross derivatives.*

Under the above assumptions, we can guarantee that there exists an in-context learning algorithm that in the limit of infinitely long contexts generates Bayes optimal predictions for all the test environments that fall in Voronoi cells of training parameters weighted by a certain vector.

Proof. The training proceeds as follows. Train an autoencoder on training data under the constraint that the output of the encoder follow a Gaussian distribution with independent components conditional on each y, e. This is stated as the following minimization.

$$\hat{g}, \hat{f}, \hat{\mu}_e^y, \hat{\Sigma}_e^y = \arg \min_{\tilde{g}, \tilde{f}, \{\mu_e^y, \Sigma_e^y\}} \mathbb{E}[\|(\tilde{g} \circ \tilde{f}(x) - x)\|^2] + \alpha \sum_{y,e} \mathsf{KL}\Big(p_{\tilde{z}}(\cdot|y,e) \parallel \mathcal{N}(\mu_e^y, \Sigma_e^y)\Big) \tag{38}$$

where $\tilde{z} = \tilde{f}(x)$, $p_{\tilde{z}}(\cdot|y,e)$ is the distribution of $\tilde{z}$. The first term is standard reconstruction loss and the second term is the KL divergence between distribution of $\tilde{z}$ and a Normal distribution with independent components. Also, estimate the class probabilities for each environment and denote them as $\hat{p}_e^y$. Similar to the proof of Theorem 3 define $\hat{\gamma}_e = [(\hat{p}_e^y, \hat{\mu}_e^y, \hat{\Sigma}_e^y)_{y \in \{0,1\}}]$

The model at test time works as follows. We first use the trained encoder $\hat{f}$ and generate $\tilde{z}$ for test time inputs. After this the model operates in exactly the same way on $\tilde{z}'s$ as in the proof of Theorem 3. Basically the output of encoder takes place of raw x's in the procedure described in proof of Theorem 3.

The assumptions in this theorem along with following i) $\tilde{z}$ follows a Gaussian distribution with independent components, ii) $g(\tilde{z})$ follows distribution of x conditional on y, e for each y, e, implies we can use the previous result in Theorem 7 to conclude that $\tilde{z} = \Lambda\Pi z + r$. Observe that $\tilde{z}$ also follows a Gaussian distribution with independent components conditional on each y, e. In the limit of infinitely long contexts, $\hat{\gamma}_e$ is equal to scaled means of original training environments and covariances also scaled componentwise according to the transform $\Lambda\Pi$. We can now apply the previous Theorem 3 on $\tilde{z}'s$ as follows. If the parameters of the test environment are in the Voronoi cell of the train distribution of $\tilde{z}'s$, then the procedure described above continues to generate Bayes optimal predictions in those environments.

□

A.6 Comparing ICRM and ERM under the lens of invariance

The label y is related to x^1 and mean of x^2 in environment e as follows.

$$y \leftarrow \alpha x^1 + \beta \mu_e^2 + \varepsilon \tag{39}$$

ERM learns a linear model on two dimensional feature vector $x = (x^1, x^2)$. The closed form solution for linear regression is $\Lambda^{-1}\rho$, where $\Lambda = \mathbb{E}[XX^\top]$, which is assumed to be invertible, and $\rho = \mathbb{E}[XY]$. The covariance matrix of X is defined as $\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}$.

Proposition 2. *Let $\mathbb{E}[X^1|E=e] = 0$ for all $e \in \mathcal{E}$. If Σ is invertible, $\beta \neq 0$, $\sigma_{12} \neq 0$, $\mu_e^2 \neq 0$ for some $e \in \mathcal{E}_{tr}$, then the coefficient estimated by ERM for x^1 is not the same as the invariant coefficient α.*

Proof. We compute ρ first.

$$\begin{aligned} \rho = \mathbb{E}[XY] &= \begin{bmatrix} \alpha\mathbb{E}[(X^1)^2] + \beta\mathbb{E}[\mu_E^2 X^1] \\ \alpha\mathbb{E}[X^1X^2] + \beta[\mu_E^2 X^2] \end{bmatrix} \\ &= \alpha \begin{bmatrix} \sigma_1^2 \\ \sigma_{12} + \frac{\beta}{\alpha}\delta \end{bmatrix}, \end{aligned} \tag{40}$$

where $\delta = \mathbb{E}[(\mu_E^2)^2]$.

Next, we compute Λ.

$$\Lambda = \mathbb{E}[XX^\top] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 + \delta \end{bmatrix}. \tag{41}$$

The solution to ERM is

$$\begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = \frac{\alpha}{(\sigma_2^2 + \delta)\sigma_1^2 - \sigma_{12}^2} \begin{bmatrix} \sigma_2^2 + \delta & -\sigma_{12} \\ -\sigma_{12} & \sigma_1^2 \end{bmatrix} \begin{bmatrix} \sigma_1^2 \\ \sigma_{12} + \frac{\beta}{\alpha}\delta \end{bmatrix}. \tag{42}$$

Simplifying the above, we obtain the coefficient for x_1 to be

$$\alpha' = \alpha - \frac{\sigma_{12}\beta\mathbb{E}[(\mu_E^2)^2]}{\sigma_1^2\big(\sigma_2^2 + \mathbb{E}[(\mu_E^2)^2]\big) - \sigma_{12}^2}. \tag{43}$$

Owing to the assumptions, $\beta \neq 0, \sigma_{12} \neq 0$ and μ_e^2 for some e we obtain that the second term in the above is not zero. As a result, the estimate computed by ERM for α is biased. □

Proposition 3. *Let $\mathbb{E}[X_1|E = e] = 0$ for all $e \in \mathcal{E}$. If Σ is invertible, $\beta \neq 0$, $\sigma_{12} \neq 0$, $\mu_e^2 \neq 0$. The error of ERM in test environment increases in σ_1^2*

Proof. The error of ERM is given as

$$\begin{aligned} &\mathbb{E}[(\alpha X^1 + \beta\mu_e^2 - \alpha' X^1 - \beta' X^2)^2] + \sigma_\varepsilon^2 \\ &= (\alpha - \alpha^{'})^2\sigma_1^2 + \beta^2\mathbb{E}[(\mu_E^2)^2] + (\beta')^2\mathbb{E}[(X^2)^2] - 2\beta\beta^{'}\mathbb{E}[(\mu_E^2)^2] - 2(\alpha - \alpha')\beta\sigma_{12} + \sigma_\varepsilon^2, \end{aligned} \tag{44}$$

where σ_ε^2 is the variance of the noise variable ε. If we take the derivative of the above error w.r.t σ_1^2, we obtain $(\alpha - \alpha')^2$, which is positive. This completes the proof.

□

ICRM learns a linear model on $(x^1, x^2, \mu_e^1, \mu_e^2)$. We study two settings to analyze the error of ICRM at test time. If at test time, the model has seen sufficiently long contexts, then it knows the means corresponding to x^1 and x^2 and the model achieves the test error of σ_ε^2. On the other hand, if the context is empty, then also note that the expected error of the model is $\beta^2\|\mu_{e'}^2\|^2$ (assuming the model uses a default value of zero for the mean in the absence of any context), where $\mu_{e'}^2$ is the mean of x^2 in environment e'. Since the error of ICRM in the absence of any context is independent of variance of x^1, the error of ERM can be much worse than that of ICRM in this setting as well.

Extending the above example beyond linear settings. Let us consider a more general setting.

$$\begin{aligned} y &= u(x^1, \mu_e^2) + \varepsilon, \\ x^2 &= v(\mu_e^2, \vartheta), \end{aligned} \tag{45}$$

where $u(\cdot)$ and $v(\cdot)$ are maps (potentially non-linear), ε and ϑ are independent zero mean noise variables. Following the same line of thought as the above example. ICRM learns a non-linear model on $(x_1, x_2, \mu_1^e, \mu_2^e)$ and learns $\mathbb{E}[Y|x^1, x^2, \mu_e^1, \mu_e^2]$. From equation 45, it follows that

$$Y \perp (X^2, \mu_E^1)|(X^1, \mu_E^2) \implies \mathbb{E}[Y|x^1, x^2, \mu_e^1, \mu_e^2] = \mathbb{E}[Y|x^1, \mu_e^2] = u(x^1, \mu_e^2).$$

From the above it follows that ICRM learns $u(x^1, \mu_e^2)$. In comparison, consider standard ERM learns a non-linear model on (x^1, x^2). Consider the DAG corresponding to setting equation 45. We assume that the joint distribution described in equation 45 is Markov w.r.t to the following DAG $X^1 \to Y \leftarrow \mu_E^2 \to X^2$. As a result, $Y \not\perp X^2|X^1$. This follows from the fact there is a path Y to X^2 through μ_E^2 and is not blocked by X^1. From $Y \not\perp X^2|X^1$ it follows that ERM learns a predictor that relies on both x^1 and x^2. Therefore, ICRM learns the right invariant model and does not rely on x^2 and ERM relies on spurious feature x^2.

A.7 Illustration of failure of existing MTL methods

In this section, we provide a simple example to show the failure mode of marginal transfer learning (MTL) methods that are based on averaging $\frac{1}{|c|}\sum_{x_i \in c} \Phi(\cdot)$ to summarize information about the environment. These methods can be summarized to take the following form:

$$f\left(\frac{1}{|c|}\sum_{x_i \in c} \Phi(x_i), x\right). \tag{46}$$

We are only going to consider maps Φ that are differentiable.

Example. Suppose we want to learn the following function

$$w(x, c) = \frac{1}{|c|}\sum_{x_i \in C} I(x < x_i), \tag{47}$$

where x_i is the i^{th} input in the context and x is the current query and $I(\cdot)$ is indicator function that takes a value of one if the argument inside is true and zero otherwise. We claim that if $f\left(\frac{1}{|c|}\sum_{x_i \in c} \Phi(x_i), x\right) = w(x, c)$ for all $x \in \mathbb{R}, c \in \mathbb{R}^{|c|}$, then the output dimension of Φ grows in context length $|c|$. Suppose this was not the case. If $\Phi' s$ output dimension is smaller than $|c|$, then Φ cannot be a differentiable bijection. As a result, there exists two contexts c and c' of same length for which $\sum_{x_i \in c} \Phi(x_i) = \sum_{x_i \in c'} \Phi(x_i)$. We argue that there exists an x such that $w(x, c) \neq w(x, c')$. This would lead to a contradiction as $f\left(\frac{1}{|c|}\sum_{x_i \in c} \Phi(x_i), x\right) = w(x, c)$ for all x, c. Without loss of generality, suppose that the smallest value of context c is smaller than that in context c'. If x is larger than smallest value of c but lesser than smallest value of c', then $w(x, c') = 1$ on the other hand $w(x, c) \leq 1 - \frac{1}{|c|}$.

We can translate the insight from the above example into more general settings. Consider any map $w(x, c)$, that satisfies the following property. For no two distinct contexts c and $c^{'}$, $w(x, c) = w(x, c')$ for all $x \in \mathbb{R}$. Maps of the form $f\left(\frac{1}{|c|}\sum_{x_i \in c} \Phi(x_i), x\right)$ can only approximate such $w' s$ provided dimension of Φ grows in length of c.

We explain how the above example can be described by attention-based architectures with much fewer parameters. First take the current query x and transform it through a linear map $\tilde{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}$ and transform the past context values through a linear map as well to obtain a transform for x_i to $\tilde{x}_i = \begin{bmatrix} 1 \\ x_i \end{bmatrix}$. We set the Query Q and Key K matrices such that $Q^\top K = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ and thus $\tilde{x}^\top Q^\top K \tilde{x}_i = (-x + x_i)$. Instead of softmax, if we pass the output through a sigmoid, we obtain $\sigma(\tau \tilde{x}^\top Q^\top K \tilde{x}_i) = \frac{1}{1+e^{-\tau(x_i - x)}}$. If τ is sufficiently large, then this approximates $I(x < x_i)$. Therefore, one layer linear attention with sigmoid and sufficiently high τ achieves the target, i.e., $\sum_{x_i \in c} \sigma(\tau \tilde{x}^\top Q^\top K \tilde{x}_i) \approx \sum_{x_i \in C} I(x < x_i)$.

B Related work

A brief tour of domain generalization. [31] developed kernel methods to learn transformations such that the distance between the feature distributions across domains is minimized and the information between the features and the target labels is preserved. The pioneering work of [15] proposes a method inspired from generative adversarial networks to learn feature representations that are similar across domains. [44] developed a method based on a natural strategy to match the means and covariances of feature representations across domains. [27] went a step further to enforce invariance on the distribution of representations conditional on the labels. In a parallel line of work, led by [35, 39, 3], the proposals sought to learn representations such that the distribution of labels conditional on the representation are invariant across domains. These works were followed by several interesting proposals

to enforce invariance – [45, 25, 2, 19, 9, 29, 24, 32, 34, 38, 50, 10, 53, 58, 12, 37, 49, 30, 51, 42, 13] – which is an incomplete representative list. See [43] for a more comprehensive survey of these works. Most of the above works have focused on learning features that enable better generalization. Recently there been an intriguing line of work from [22, 18] that shifts the focus from feature learning to last layer retraining. These works show that under certain conditions (e.g., avaiability of some data that does not carry spurious correlations) one can carry out last layer retraining and achieve significant out-of-distribution performance improvements.

In the main body of the paper, we already discussed the other prominent line of work in domain generalization on marginal transfer learning, where the focus is to leverage the distributional features and learn environment specific relationships. This line of work was started by the notable work of [7] and has been followed up by several important proposals such as [57, 5].

C Supplementary experimental details and assets disclosure

C.1 Assets

We do not introduce new data in the course of this work. Instead, we use publicly available widely used image datasets for the purposes of benchmarking and comparison.

C.2 Hardware and setup

Each experiment was performed on 8 NVIDIA Tesla V100 GPUs with 32GB accelerator RAM for a single training run. The CPUs used were Intel Xeon E5-2698 v4 processors with 20 cores and 384GB RAM. All experiments use the PyTorch deep-learning framework

C.3 Datasets

C.3.1 Federated Extended MNIST (FEMNIST)

Building on the Extended MNIST (EMNIST) dataset, which includes images of handwritten uppercase and lowercase alphabets along with digits, FEMNIST [57] enriches this data by attributing each data point to its originating writer. This extension associates each 28×28-sized image in the dataset to one of the 62 classes. In our setup, each writer serves as a distinct environment. We evaluate the performance of each method based on both worst-case and average accuracy across a set of 35 test users, who are distinct from the 262 training users and 50 validation users. Unlabelled data from an environment in this dataset could provide cues about the writing style of the user and disambiguate data points.

C.3.2 Rotated MNIST

We employ a customized version of the MNIST dataset as in [57]. The dataset contains images rotated in increments of 10 degrees, ranging from 0 to 130 degrees. Each degree of rotation constitutes a separate environment, effectively acting as a distinct value. The training set for the two most extreme rotations, 120 and 130 degrees, contains only 108 data points each. For rotations between 90 and 110 degrees, each environment includes 324 data points. The total training set comprises 32,292 points. For evaluation, test images are generated from the MNIST test set, and are duplicated for each environment. Performance metrics include both worst-case and average accuracy across these testing domains. Analogous to FEMNIST, unlabeled samples from an environment within this dataset can assist in distinguishing images that may seem similar due to their rotated orientations.

C.3.3 WILDS Camelyon17

We use the Camelyon17 dataset, part of the WILDS benchmark [23], which features image patches derived from whole-slide lymph node sections of patients with potential metastatic breast cancer. Each patch is labeled to indicate the presence or absence of a tumor. In our experimental design, each participating hospital is treated as a distinct environment. The dataset is partitioned in alignment with

the official WILDS configuration: three hospitals contribute to the training set, a fourth is designated for validation, and the remaining hospital's data is used for testing.

C.3.4 Tiny ImageNet-C

Adapting the methodology from [17], we introduce 56 distinct distortions to the training set, treating each as a separate environment. For evaluation, we use a non-overlapping set of 22 test distortions, largely differing in nature from those used in training. Each 64×64-sized distorted image is associated with one of the 200 classes in the dataset. This setup permits an investigation into whether exposure to distortions during training equips the model to better manage novel distortions during testing. We assess performance through both worst-case and average accuracies across these test distortions.

C.4 Experimental protocols

To ensure a fair comparison across different algorithms for each dataset, we use a standardized neural network backbone. The details for these architectures are provided in Table 3 and Table 4. We use the ConvNet architecture as outlined in [57].

For ICRM, the same backbone is used to featurize the input, which is then processed by the decoder-only Transformer [48] architecture from the GPT-2 Transformer family [36]. Our model is standardized to have 12 layers, 4 attention heads, and a 128-dimensional embedding space across all datasets. Linear layers are employed to map both the input sequence to the transformer's latent embedding and the model's predicted output vector to the output label. For training ICRM on larger datasets like WILDS Camelyon17 and Tiny ImageNet-C, we start with a ResNet50 model pre-trained on ImageNet (as shown in Table 3) and freeze all batch normalization layers before fine-tuning.

We adopt the same Context Network as used in ARM, specifically retaining their choice of output channels – one for smaller datasets like FEMNIST and Rotated MNIST, and three for the others.

For TENT, all reported metrics are based on its episodic version, where the model is reset to its trained state after processing each batch. This ensures a fair comparison with other methods. Additionally, during testing, the model's parameters are updated for 10 steps using stochastic gradient descent by minimization test entropy across all datasets.

Table 3: Network architectures for each dataset.

Dataset	Architecture	
	ICRM	Others
FEMNIST Rotated MNIST	ConvNet + GPT2 Transformer	ConvNet
Camelyon17 Tiny ImageNet-C	ResNet-50 + GPT2 Transformer	ResNet-50

Table 4: ConvNet architecture for [57]. We use 2×2 kernels and "same" padding.

#	**Layer**
1	Conv2D (in=d, out=128)
2	BatchNorm2d (dim=129)
3	ReLU
4	Max Pooling (2)
5	Conv2D (in=128, out=128)
6	BatchNorm2d (dim=128)
7	ReLU
8	Max Pooling (2)
9	Global average-pooling

We list all hyperparameters, their default settings, and search boundaries for random sweeps in Table 5. The maximum context length, or support, is fixed at 100 for all algorithms. All models are optimized using the Adam optimizer [21]. To ensure a fair comparison, we perform a random search of 5 trials across the hyperparameter range (refer to Table 5) for each algorithm. The model with the highest validation set accuracy is selected for each run. We then report the average of this number across three independent runs of the entire sweep, and its corresponding standard error.

Table 5: Hyperparameters, their default values and distributions for random search.

Condition	Parameter	Default value	Random distribution
ResNet	learning rate	0.0001	$10^{\text{Uniform}(-5,-3.5)}$
	weight decay	0	$10^{\text{Uniform}(-6,-2)}$
not ResNet	learning rate	0.0001	$10^{\text{Uniform}(-4.5,-2.5)}$
	weight decay	0	$10^{\text{Uniform}(-6,-2)}$

D Additional experiments

D.1 Adaptation curves of various algorithms

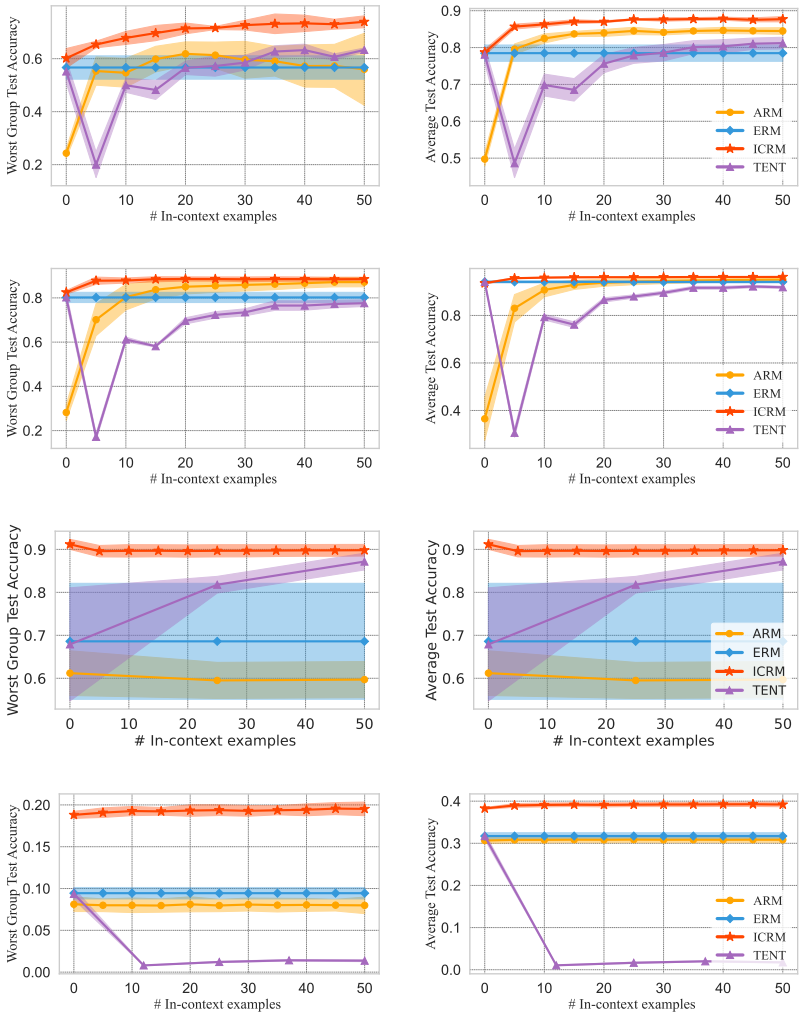


Figure 4: Accuracy adaptation curves for worst accuracy (left) and average accuracy (right) across the test environment as a function of increasing count of context samples. Showing results in order for FEMNIST(top), RotatedMNIST, WILDS Camelyon17 and Tiny ImageNet-C(bottom). The average and worst-case accuracy plots for WILDS Camelyon17 are identical since the dataset contains only a single test environment.

D.2 Domain generalization accuracies per algorithm and dataset

D.2.1 Adaptation to distribution shift

In our experiments, we compare ICRM against marginal transfer methods such as Adaptive Risk Minimization [57, ARM], test-time adaptation proposals such as TENT [52] and Empirical Risk Minimization [47, ERM]. Table 6 and Table 7 demonstrate the average and worst group out-of-distribution performance, respectively, accompanied by the corresponding standard errors. These statistics are computed across three independent runs of the entire sweep, wherein the model selected for evaluation is the one with hyper-parameters yielding the highest validation accuracy.

Table 6: Average out-of-distribution test accuracies along with their corresponding standard errors for various counts of context samples. The methods compared include Adaptive Risk Minimization (ARM), Empirical Risk Minimization (ERM), Test Entropy Minimization (TENT), and our method ICRM on FEMNIST, Rotated MNIST, WILDS Camelyon17 and Tiny-ImageNet-C.

Dataset / algorithm	**Average test accuracy (by # in-context examples)**				
FEMNIST	0	25	50	75	100
ARM	49.5 ± 1.0	83.9 ± 0.5	84.4 ± 0.5	84.7 ± 0.6	84.6 ± 0.3
TENT	78.1 ± 1.2	77.9 ± 1.2	81.2 ± 0.9	82.5 ± 0.9	83.3 ± 0.8
ERM	79.3 ± 0.4	79.3 ± 0.4	79.3 ± 0.4	79.3 ± 0.4	79.3 ± 0.4
ICRM	78.7 ± 0.5	87.2 ± 0.4	87.4 ± 0.5	87.5 ± 0.2	87.8 ± 0.2
Rotated MNIST	0	25	50	75	100
ARM	36.5 ± 5.2	94.2 ± 0.7	95.1 ± 0.4	95.3 ± 0.4	95.5 ± 0.3
TENT	94.1 ± 0.3	88.0 ± 0.4	91.9 ± 0.3	93.8 ± 0.2	94.3 ± 0.2
ERM	94.2 ± 0.3	94.2 ± 0.3	94.2 ± 0.3	94.2 ± 0.3	94.2 ± 0.3
ICRM	93.6 ± 0.2	96.1 ± 0.1	96.2 ± 0.1	96.2 ± 0.1	96.2 ± 0.1
WILDS Camelyon17	0	25	50	75	100
ARM	61.2 ± 5.2	59.5 ± 4.2	59.7 ± 4.2	59.7 ± 4.3	59.7 ± 4.2
TENT	67.9 ± 7.6	81.8 ± 1.1	87.2 ± 1.1	89.4 ± 1.1	89.4 ± 1.0
ERM	68.6 ± 7.8	68.6 ± 7.8	68.6 ± 7.8	68.6 ± 7.8	68.6 ± 7.8
ICRM	92.0 ± 0.6	90.7 ± 0.8	90.8 ± 0.8	90.8 ± 0.8	90.8 ± 0.8
Tiny ImageNet-C	0	25	50	75	100
ARM	30.8 ± 0.2	31.0 ± 0.2	31.0 ± 0.2	31.0 ± 0.2	31.0 ± 0.2
TENT	31.7 ± 0.5	1.6 ± 0.1	1.7 ± 0.1	2.0 ± 0.1	2.1 ± 0.1
ERM	31.8 ± 0.6	31.8 ± 0.6	31.8 ± 0.6	31.8 ± 0.6	31.8 ± 0.6
ICRM	38.3 ± 0.1	39.2 ± 0.3	39.2 ± 0.3	39.2 ± 0.3	39.2 ± 0.3

Table 7: Worst environment out-of-distribution test accuracies along with their corresponding standard errors for various counts of context samples. The methods compared include Adaptive Risk Minimization (ARM), Empirical Risk Minimization (ERM), Test Entropy Minimization (TENT), and our method ICRM on FEMNIST, Rotated MNIST, WILDS Camelyon17 and Tiny-ImageNet-C.

Dataset / algorithm	**Worst case test accuracy (by # in-context examples)**				
FEMNIST	0	25	50	75	100
ARM	23.6 ± 1.7	59.5 ± 3.5	60.7 ± 3.8	57.0 ± 7.3	58.8 ± 4.0
TENT	55.2 ± 2.5	57.2 ± 2.2	63.3 ± 0.4	65.9 ± 0.6	67.2 ± 1.0
ERM	59.0 ± 0.2	59.0 ± 0.2	59.0 ± 0.2	59.0 ± 0.2	59.0 ± 0.2
ICRM	59.8 ± 0.7	69.3 ± 0.0	70.6 ± 2.3	70.6 ± 1.5	70.6 ± 0.7
Rotated MNIST	0	25	50	75	100
ARM	28.2 ± 2.1	85.3 ± 1.6	87.2 ± 1.0	87.9 ± 1.0	87.9 ± 0.9
TENT	80.2 ± 1.3	88.5 ± 0.8	88.5 ± 0.9	80.2 ± 1.0	81.3 ± 1.0
ERM	80.8 ± 1.1	80.8 ± 1.1	80.8 ± 1.1	80.8 ± 1.1	80.8 ± 1.1
ICRM	82.5 ± 0.5	88.5 ± 0.5	88.5 ± 0.5	88.8 ± 0.5	88.8 ± 0.4
WILDS Camelyon17	0	25	50	75	100
ARM	61.2 ± 5.2	59.5 ± 4.2	59.7 ± 4.2	59.7 ± 4.3	59.7 ± 4.2
TENT	67.9 ± 7.6	81.8 ± 1.1	87.2 ± 1.1	89.4 ± 1.1	89.4 ± 1.0
ERM	68.6 ± 7.8	68.6 ± 7.8	68.6 ± 7.8	68.6 ± 7.8	68.6 ± 7.8
ICRM	92.0 ± 0.6	90.7 ± 0.8	90.8 ± 0.8	90.8 ± 0.8	90.8 ± 0.8
Tiny ImageNet-C	0	25	50	75	100
ARM	$8.2 \pm 0{,}3$	8.3 ± 0.3	8.2 ± 0.3	8.3 ± 0.3	8.2 ± 0.3
TENT	1.2 ± 0.4	1.4 ± 0.0	1.6 ± 0.1	1.6 ± 0.0	1.6 ± 0.0
ERM	9.5 ± 0.4	9.5 ± 0.4	9.5 ± 0.4	9.5 ± 0.4	9.5 ± 0.4
ICRM	18.8 ± 0.2	19.2 ± 0.1	19.5 ± 0.2	19.5 ± 0.1	19.4 ± 0.2

D.2.2 Robustness of ICRM in the absence of environment labels

As outlined in Section 4, the training regimen of ICRM assumes a dataset $\mathcal{D} = \{(x_i, y_i, e_i)\}_{i=1}^n$ collected under multiple training environments $e_i \in \mathcal{E}_{\text{tr}}$. However, in scenarios lacking such domain separation during training, does ICRM continue to show an edge over ERM baselines? To study this question, we modify the sampling strategy: rather than constructing context vectors containing examples from one environment, we construct context vectors containing iid samples from all of the environments pooled together. To continue to test for out-of-distribution generalization, however, we evaluate the performance on examples from a novel test environment. We term this modified approach ICRM-Mix.

Table 8 and Table 9 contrasts the performance of ICRM with ICRM-Mix. ICRM consistently outperforms ICRM-Mix across varying counts of in-context samples on both FEMNIST and Rotated MNIST. Surprisingly, ICRM-Mix and ICRM perform similarly on WILDS Camelyon17 and Tiny ImageNet-C. Consider a setting where the model benefits the most attending to examples from the same class or related classes. If classes are distributed uniformly across domains, then ICRM and ICRM-mix are bound to perform similarly. Consider another setting where the model benefits the most by attending to environment-specific examples such as characters drawn by the same user. In such a case, ICRM and ICRM-mix have very different performances.

Table 8: Average out-of-distribution test accuracies along with their corresponding standard errors for ICRM and ICRM-Mix across FEMNIST, Rotated MNIST, WILDS Camelyon17 and Tiny-ImageNet-C. ICRM-Mix trains on sequences with samples drawn i.i.d. from the unified dataset comprising various environments.

Dataset / algorithm	**Average test accuracy (by # in-context examples)**				
FEMNIST	0	25	50	75	100
ICRM	78.7 ± 0.5	87.2 ± 0.4	87.4 ± 0.5	87.5 ± 0.2	87.8 ± 0.2
ICRM-Mix	77.6 ± 0.8	81.1 ± 0.2	81.1 ± 0.2	80.9 ± 0.3	80.9 ± 0.1
Rotated MNIST	0	25	50	75	100
ICRM	93.6 ± 0.2	96.1 ± 0.1	96.2 ± 0.1	96.2 ± 0.1	96.2 ± 0.1
ICRM-Mix	88.9 ± 1.4	92.6 ± 0.3	92.7 ± 0.2	92.6 ± 0.3	92.7 ± 0.2
WILDS Camelyon17	0	25	50	75	100
ICRM	92.0 ± 0.6	90.7 ± 0.8	90.8 ± 0.8	90.8 ± 0.8	90.8 ± 0.8
ICRM-Mix	92.9 ± 0.3	90.7 ± 0.6	90.8 ± 0.5	90.7 ± 0.5	90.7 ± 0.5
Tiny ImageNet-C	0	25	50	75	100
ICRM	38.3 ± 0.1	39.2 ± 0.3	39.2 ± 0.3	39.2 ± 0.3	39.2 ± 0.3
ICRM-Mix	38.4 ± 0.2	39.3 ± 0.2	39.3 ± 0.2	39.3 ± 0.2	39.3 ± 0.2

D.2.3 Understanding the impact of architecture

Table 10 and Table 11 demonstrate the average and worst group out-of-distribution performance of these approaches, respectively, along with the corresponding standard errors. These statistics are computed across three independent runs of the entire sweep, wherein the model selected for evaluation is the one with hyper-parameters yielding the highest validation accuracy.

Table 9: Worst environment out-of-distribution test accuracies along with their corresponding standard errors for ICRM and ICRM-Mix across FEMNIST, Rotated MNIST, WILDS Camelyon17 and Tiny-ImageNet-C. ICRM-Mix trains on sequences with samples drawn i.i.d. from the unified dataset comprising various environments.

Dataset / algorithm	**Worst case test accuracy (by # in-context examples)**				
FEMNIST	0	25	50	75	100
ICRM	59.8 ± 0.7	69.3 ± 0.0	70.6 ± 2.3	70.6 ± 1.5	70.6 ± 0.7
ICRM-Mix	57.5 ± 1.4	62.7 ± 1.1	65.0 ± 0.3	64.1 ± 1.5	62.9 ± 2.3
Rotated MNIST	0	25	50	75	100
ICRM	82.5 ± 0.5	88.5 ± 0.5	88.5 ± 0.5	88.8 ± 0.5	88.8 ± 0.4
ICRM-Mix	68.8 ± 3.8	77.1 ± 0.7	76.8 ± 0.9	76.4 ± 0.9	76.6 ± 0.9
WILDS Camelyon17	0	25	50	75	100
ICRM	92.0 ± 0.6	90.7 ± 0.8	90.8 ± 0.8	90.8 ± 0.8	90.8 ± 0.8
ICRM-Mix	92.9 ± 0.3	90.7 ± 0.6	90.8 ± 0.5	90.7 ± 0.5	90.7 ± 0.5
Tiny ImageNet-C	0	25	50	75	100
ICRM	18.8 ± 0.2	19.2 ± 0.1	19.5 ± 0.2	19.5 ± 0.1	19.4 ± 0.2
ICRM-Mix	18.7 ± 0.2	19.2 ± 0.2	19.4 ± 0.1	19.5 ± 0.1	19.4 ± 0.1

Table 10: Average out-of-distribution test accuracies along with their corresponding standard errors for ARM$^+$ and ERM$^+$ in contrast to their base algorithms, ARM and ERM across FEMNIST, Rotated MNIST, WILDS Camelyon17 and Tiny-ImageNet-C.

Dataset / algorithm	**Average test accuracy (by # in-context examples)**				
FEMNIST	0	25	50	75	100
ARM	49.5 ± 1.0	83.9 ± 0.5	84.4 ± 0.5	84.7 ± 0.6	84.6 ± 0.3
ARM$^+$	71.4 ± 1.2	83.4 ± 0.2	84.0 ± 0.2	83.8 ± 0.2	83.5 ± 0.1
ERM	79.3 ± 0.4	79.3 ± 0.4	79.3 ± 0.4	79.3 ± 0.4	79.3 ± 0.4
ERM$^+$	77.4 ± 1.3	77.4 ± 1.3	77.4 ± 1.3	77.4 ± 1.3	77.4 ± 1.3
Rotated MNIST	0	25	50	75	100
ARM	36.5 ± 5.2	94.2 ± 0.7	95.1 ± 0.4	95.3 ± 0.4	95.5 ± 0.3
ARM$^+$	86.9 ± 2.0	92.6 ± 0.7	92.7 ± 0.6	92.8 ± 0.6	92.8 ± 0.6
ERM	94.2 ± 0.3	94.2 ± 0.3	94.2 ± 0.3	94.2 ± 0.3	94.2 ± 0.3
ERM$^+$	94.3 ± 0.4	94.3 ± 0.4	94.3 ± 0.4	94.3 ± 0.4	94.3 ± 0.4
WILDS Camelyon17	0	25	50	75	100
ARM	61.2 ± 5.2	59.5 ± 4.2	59.7 ± 4.2	59.7 ± 4.3	59.7 ± 4.2
ARM$^+$	55.8 ± 0.8	55.1 ± 1.7	55.0 ± 1.7	55.0 ± 1.8	55.0 ± 1.8
ERM	68.6 ± 7.8	68.6 ± 7.8	68.6 ± 7.8	68.6 ± 7.8	68.6 ± 7.8
ERM$^+$	50.1 ± 0.1	50.1 ± 0.1	50.1 ± 0.1	50.1 ± 0.1	50.1 ± 0.1
Tiny ImageNet-C	0	25	50	75	100
ARM	30.8 ± 0.2	31.0 ± 0.2	31.0 ± 0.2	31.0 ± 0.2	31.0 ± 0.2
ARM$^+$	5.5 ± 0.2	5.7 ± 0.2	5.7 ± 0.2	5.7 ± 0.2	5.7 ± 0.2
ERM	31.8 ± 0.6	31.8 ± 0.6	31.8 ± 0.6	31.8 ± 0.6	31.8 ± 0.6
ERM$^+$	29.7 ± 0.3	29.7 ± 0.3	29.7 ± 0.3	29.7 ± 0.3	29.7 ± 0.3

Table 11: Worst environment out-of-distribution test accuracies along with their corresponding standard errors for ARM$^+$ and ERM$^+$ in contrast to their base algorithms, ARM and ERM across FEMNIST, Rotated MNIST, WILDS Camelyon17 and Tiny-ImageNet-C.

Dataset / algorithm	**Worst case test accuracy (by # in-context examples)**				
FEMNIST	0	25	50	75	100
ARM	23.6 ± 1.7	59.5 ± 3.5	60.7 ± 3.8	57.0 ± 7.3	58.8 ± 4.0
ARM$^+$	51.7 ± 2.2	63.0 ± 2.1	64.0 ± 0.8	60.7 ± 1.6	62.0 ± 0.8
ERM	59.0 ± 0.2	59.0 ± 0.2	59.0 ± 0.2	59.0 ± 0.2	59.0 ± 0.2
ERM$^+$	53.3 ± 2.7	53.3 ± 2.7	53.3 ± 2.7	53.3 ± 2.7	53.3 ± 2.7
Rotated MNIST	0	25	50	75	100
ARM	28.2 ± 2.1	85.3 ± 1.6	87.2 ± 1.0	87.9 ± 1.0	87.9 ± 0.9
ARM$^+$	71.4 ± 2.6	80.9 ± 1.8	81.0 ± 1.8	81.2 ± 1.9	81.1 ± 1.8
ERM	80.8 ± 1.1	80.8 ± 1.1	80.8 ± 1.1	80.8 ± 1.1	80.8 ± 1.1
ERM$^+$	81.9 ± 0.7	81.9 ± 0.7	81.9 ± 0.7	81.9 ± 0.7	81.9 ± 0.7
WILDS Camelyon17	0	25	50	75	100
ARM	61.2 ± 5.2	59.5 ± 4.2	59.7 ± 4.2	59.7 ± 4.3	59.7 ± 4.2
ARM$^+$	55.8 ± 0.8	55.1 ± 1.7	55.0 ± 1.7	55.0 ± 1.8	55.0 ± 1.8
ERM	68.6 ± 7.8	68.6 ± 7.8	68.6 ± 7.8	68.6 ± 7.8	68.6 ± 7.8
ERM$^+$	50.1 ± 0.1	50.1 ± 0.1	50.1 ± 0.1	50.1 ± 0.1	50.1 ± 0.1
Tiny ImageNet-C	0	25	50	75	100
ARM	8.2 ± 0.3	8.3 ± 0.3	8.2 ± 0.3	8.3 ± 0.3	8.2 ± 0.3
ARM$^+$	1.9 ± 0.1	1.9 ± 0.1	1.9 ± 0.1	1.9 ± 0.1	1.9 ± 0.1
ERM	9.5 ± 0.4	9.5 ± 0.4	9.5 ± 0.4	9.5 ± 0.4	9.5 ± 0.4
ERM$^+$	8.3 ± 0.3	8.3 ± 0.3	8.3 ± 0.3	8.3 ± 0.3	8.3 ± 0.3

D.3 Investigating attention in ICRM

As discussed in Section 2, a special feature of ICRM is its ability to learn an amortization function by paying attention to the input query and its context. To better understand this nuanced functionality, we construct a random sequence of data from the test environment and examine the attention scores between each example in this context and a novel input query across different heads of ICRM. Figure 5 illustrates attention scores from a single head for two query images (marked in blue) for FEMNIST and Tiny ImageNet-C. The top row reveals that the model selectively attends to images featuring at least two curved arcs (marked in green) while paying little attention to a partial circle (highlighted in red). Additionally, when the query image resembles a 90-degree clockwise rotated digit "2", the model extends its attention to other augmentations of "2" within the prompt. Such attention patterns emerge solely from unlabeled examples from unseen domains, underscoring the power of amortization. Similarly, in the second row, attention is predominantly allocated to lines of length similar to that of the query (also in green), thereby disregarding shorter lines (shown in red). The third row in Figure 5 shows that the model, when presented with a query image of a train, attends not only on other trains but also on a bus—indicating a semantic understanding of similarity. In the last row, the model shows the capability to discern *individuals* across samples within the sequence.

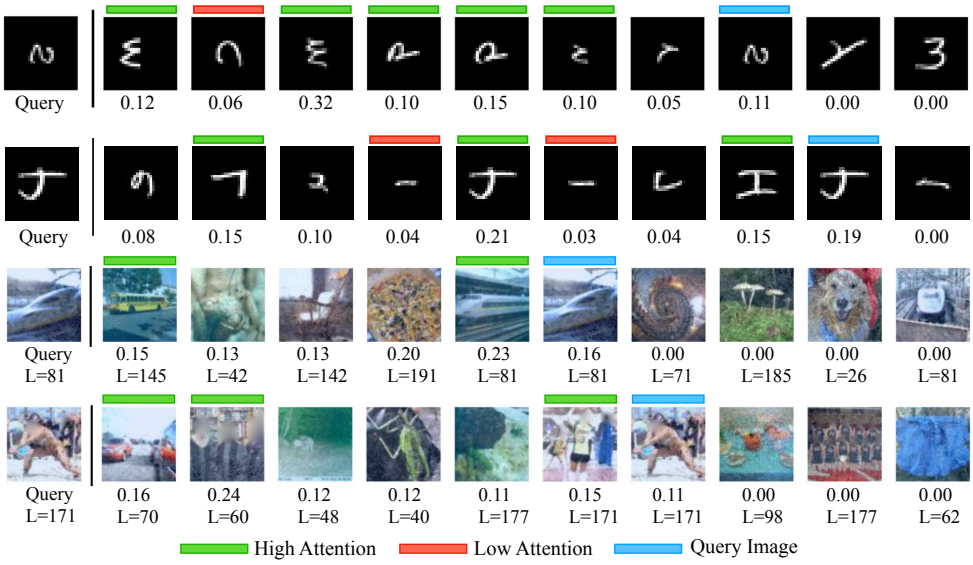


Figure 5: Attention scores for random test sequences, for ICRM on FEMNIST (top two rows) and Tiny ImageNet-C (bottom two rows). 'L' denotes the label of a given image.