ADVERSARIAL INFORMATION SEPARATION FOR DATA-SCARCE GENERALIZATION UNDER NOISY OBSERVATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning from scarce data is difficult, e.g., in scientific domains where task-relevant information is often confounded with measurement noise and experimental artifacts. Compression-based regularization like the variational information bottleneck (VIB) improves generalization by compressing representations uniformly, but may suppress weak predictive information along with noise. In contrast, adversarial separation methods preserve such information, but require explicit separation labels. To address this challenge, we propose the Adversarial Information Separation Framework (AdverISF), which learns primary and residual representations from task supervision alone. It uses a task-guided adversarial separation mechanism to reduce the statistical dependence between the representations and a multi-layer separation architecture in which subsequent layers process residual representations to extract predictive information missed by earlier layers. Experiments show that AdverISF outperforms state-of-the-art methods under severe data scarcity. On a real materials design task, it generalizes to formulations independently prepared in a subsequent batch.

1 INTRODUCTION

Generalization, the ability to predict accurately beyond the training data, determines whether a learned model is useful in practice (Goodfellow et al., 2016). A well-generalized model distinguishes task-relevant patterns (*features*) from variation introduced by measurement and experimentation (*noise*), which keeps its predictions consistent under distribution shift (Xu & Mannor, 2012). This property is critical in scientific and engineering domains (Wang et al., 2023; Liu et al., 2021), where the objective is to handle future observations rather than merely fit training distributions.

Deep learning has achieved success across diverse tasks, but its reliance on large-scale training data limits progress in data-scarce applications. Few-shot learning addresses this limitation using a handful of labeled examples and auxiliary knowledge beyond the target dataset (Snell et al., 2017). Large-scale pre-training draws on external datasets (Radford et al., 2021), whereas meta-learning extracts transferable strategies from related tasks (Finn et al., 2017). These approaches have proven effective in data-rich fields such as computer vision and natural language processing (Devlin et al., 2019; Deng et al., 2009).

However, the auxiliary data and task diversity required by these approaches are scarce in specialized domains such as materials science (Xu et al., 2023; Dou et al., 2023). In cementitious materials, data originating from different studies exhibits substantial variation in composition and physicochemical characteristics due to diverse raw material sources and complex reaction mechanisms (Li et al., 2022b; Juenger et al., 2019). These disparities hinder the integration of multi-source data into high-quality benchmarks, the pre-training of general-purpose models, and the construction of heterogeneous task collections for meta-learning.

In the absence of large-scale data, the information bottleneck (IB) theory (Tishby et al., 1999; Kawaguchi et al., 2023; Ahuja et al., 2021) offers a principled framework for learning compact predictive representations. Following the principle of minimal sufficient statistics, IB seeks an optimal representation Z that maximizes information about the target Y while minimizing the information retained about the input X. This trade-off balances representational compactness and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

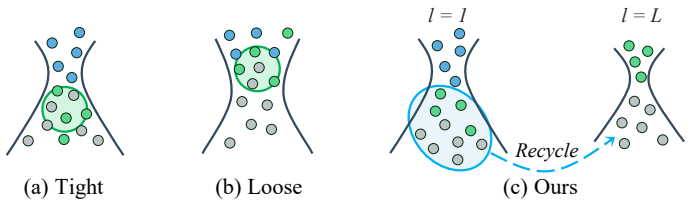

Figure 1: (a) A tight bottleneck retains dominant features (blue) but may discard subtle features (green). (b) A loose bottleneck captures subtle features but retains noise (gray). (c) Our method uses a layer-wise mechanism to recycle subtle predictive information that may otherwise be discarded.

predictive completeness. The objective is formalized as:

$$\min \mathcal{L}_{IB} = I(X;Z) - \beta I(Y;Z), \tag{1}$$

where β is a Lagrange multiplier balancing compression and prediction. While theoretically elegant, the direct application of IB to deep neural networks has been impeded by the intractability of computing mutual information in complex high-dimensional spaces (Yu et al., 2021; Alemi et al., 2017; Gowri et al., 2024; Chow & Huang, 2005).

The variational information bottleneck (VIB) (Alemi et al., 2017) addresses this intractability by employing variational inference to derive tractable variational bounds. VIB commonly uses the KL divergence between the encoder distribution and a fixed prior to enforce compression (Choi & Lee, 2024; Amjad & Geiger, 2020; Pan et al., 2021; Wang & Zhang, 2024; Hu et al., 2024). However, because this compression is applied uniformly across the representation, it may suppress weak yet predictive information together with redundancy. Under strong compression, the resulting loss of predictive completeness may limit generalization performance, as visualized in Figure 1.

Xie & Li (2024) address this loss of predictive information by introducing an information retention framework. They argue that features conventionally deemed redundant can be crucial for out-of-distribution (OOD) generalization. Yet their approach remains grounded in the VIB paradigm and relies on statistical constraints to recover weak yet predictive information. Consequently, preserving weak yet predictive information under strong compression remains challenging.

Adversarial learning offers a complement to statistical compression. By employing discriminators to enforce distribution matching, adversarial methods can separate feature components. Previous works in domain adaptation (Ganin et al., 2016; Pei et al., 2018; Long et al., 2018) and robust learning (Kim et al., 2021; Ilyas et al., 2019; Allen-Zhu & Li, 2022) have demonstrated that task-discriminative features can be separated from nuisances through adversarial min-max games.

However, these strategies commonly rely on additional supervision, such as domain labels or robust annotations, to define the separation boundary (Liang et al., 2020; Mancini et al., 2019; Zhao et al., 2019). In specialized domains with limited data, such auxiliary labels are typically unavailable, as datasets are often collected under a single experimental setting and lack the multi-domain annotations that conventional adversarial domain adaptation requires. The standard supervision-based adversarial strategy is therefore not directly applicable, even when ordinary task labels remain available.

This raises our central question. *How can we leverage adversarial mechanisms to learn task-discriminative features without explicit separation labels, thereby enhancing generalization when training data is scarce?*

Contribution To address this question, we propose the Adversarial Information Separation Framework (AdverISF), which employs dual-channel encoders to map inputs into distinct latent spaces for primary and residual representations. The main contributions of this work are as follows.

1. We propose a task-guided adversarial separation mechanism that reduces statistical dependence between the primary and residual representations through a discriminator that distinguishes their joint distribution from the product of their marginals. Unlike prior adversarial methods, our approach separates the primary and residual representations in single-domain scenarios without explicit separation labels.
2. We introduce a multi-layer separation architecture that cascades separation blocks and recycles the output of each residual branch to progressively recover complementary predictive information not

captured by earlier blocks. This design addresses a limitation of uniform compression in existing information bottleneck methods, which may suppress weak yet predictive information.

3. Extensive experiments across synthetic benchmarks and materials science applications demonstrate improvements under data-scarce conditions. On a real-world composite cement design task, AdverISF achieves an R^2 of 0.897 in predicting compressive strength on independently prepared formulations from a subsequent experimental batch, outperforming the evaluated baselines.

2 Related Work

Learning from Multiple Tasks and Domains Strategies for improving generalization often leverage external knowledge. Representative approaches include large-scale pre-training (Radford et al., 2021), meta-learning (Finn et al., 2017), and domain-specific models (Lin et al., 2023; Deng et al., 2009). However, these methods rely on large-scale datasets, sufficiently diverse and relevant training tasks, or multiple source domains. They are often inapplicable to specialized scientific fields where data is inherently scarce and high-quality auxiliary datasets are largely unavailable.

Information Bottleneck Methods The information bottleneck (IB) principle trades off prediction accuracy against compression (Tishby et al., 1999). VIB applies this to deep networks (Alemi et al., 2017), with subsequent extensions for robustness and domain adaptation (Kuang et al., 2023; Li et al., 2022a). Despite these advancements, most IB methods rely on passive statistical regularization against a fixed prior. They apply uniform compression to the representation and lack a dedicated pathway to preserve weak yet predictive information, a limitation that becomes acute in complex single-domain scenarios.

Factorized Representation Learning Disentangled representation learning (Chen et al., 2018; Hsu et al., 2024; 2023; Zhu et al., 2021), such as β-VAE (Higgins et al., 2017) and FactorVAE (Kim & Mnih, 2018), enforces factorization to enhance interpretability. However, these objectives are typically unsupervised and task-agnostic. They treat all factors equally and lack a mechanism to differentiate task-relevant features from noise. This limits their effectiveness in supervised tasks that require prioritizing task-relevant patterns.

Adversarial learning provides a flexible way to impose structural constraints on representations. Methods such as ANICA (Brakel & Bengio, 2017) employ discriminators to enforce statistical independence between features. While effective for disentanglement tasks, these methods operate in an unsupervised manner and cannot identify task relevance. Meanwhile, domain adaptation methods (Pei et al., 2018; Long et al., 2018) such as DANN (Ganin et al., 2016) use adversarial training to explicitly separate discriminative features from variations. However, these domain adaptation approaches depend on explicit supervision such as domain labels.

Challenge A critical gap remains in specialized, data-scarce domains. External knowledge approaches require auxiliary data that is scarce, while domain adaptation methods depend on explicit domain labels. Moreover, existing methods based on statistical independence, such as ANICA, apply the same pressure to every latent factor and fail to prioritize task-relevant patterns over measurement-induced variation. The fundamental challenge is to learn primary and residual representations from task supervision alone, without explicit separation labels.

3 Methodology

The Adversarial Information Separation Framework (AdverISF) treats information separation as an active adversarial game between two competing objectives. Rather than relying on passive compression via statistical regularization, we explicitly model the competition between primary representation preservation and residual information isolation. We formalize this framework in the following stages. Section 3.1 establishes the noisy observation setting, where observed inputs x confound task-relevant features z_{task} with nuisance factors z_{noise}. Section 3.2 and Section 3.3 present a dual-branch architecture that explicitly learns the primary and residual representations as z_p and z_r. The key innovation lies in an adversarial mechanism that minimizes mutual information $I(z_p; z_r)$ through a minimax game. Section 3.4 then demonstrates how this approach scales to multi-layer hierarchies for progressive feature refinement.

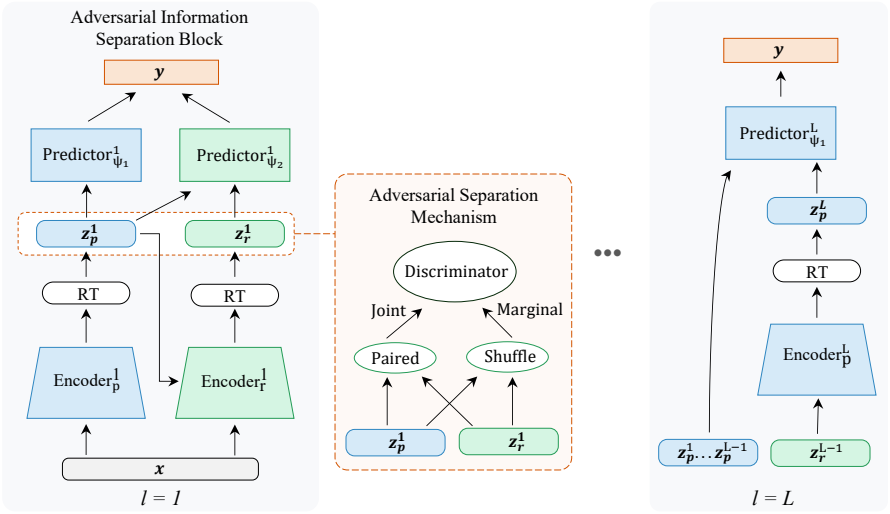

Figure 2: Overview of AdverISF. The multi-layer architecture uses dual encoders to decompose each input into primary and residual representations. An adversarial separation mechanism reduces their statistical dependence, while the residual representation is passed to the next layer to extract complementary predictive information. RT denotes the reparameterization trick.

3.1 Problem Formulation

In standard supervised learning, a model learns a mapping from the observed input $x \in \mathcal{X}$ to the target $y \in \mathcal{Y}$, typically written as $y = f(x) + \epsilon$, where ϵ denotes irreducible error. This formulation assumes that x directly encodes the task-relevant information.

In many real-world scenarios, however, the target depends on latent task-relevant features z_{task} that are not directly observable. Instead, the input x is obtained through a potentially complex observation process, while the target can be written as $y = f(z_{task}) + \epsilon$. In practical domains, the observation process may be affected by environmental interference, instrumental noise, and other confounding factors. We therefore model the observed input as

$$x = \mathcal{G}(z_{task}, z_{noise}), \tag{2}$$

where $\mathcal{G}(\cdot)$ denotes the observation mechanism and z_{noise} represents nuisance variation that does not provide stable predictive information about y. This confounding poses a key challenge for learning. Models trained on x may inadvertently rely on spurious correlations with z_{noise}, leading to poor generalization when the nuisance distribution changes.

Given this noisy observation setting, the information bottleneck (IB) framework aims to learn a compressed representation z that preserves information about the target y while reducing the information about the input x. This objective does not explicitly distinguish predictive components according to their contribution to generalization. In practice, IB methods such as the variational information bottleneck (VIB) implement this trade-off through a global statistical penalty, specifically by minimizing the KL divergence between the encoder distribution and a fixed prior.

However, in complex domains, the contributions of different predictive components are rarely uniform. Predictive information in x often exhibits a hierarchy of importance, with some components contributing more substantially to the prediction than others. We can conceptually decompose the task-relevant information into two types of components. The first type, denoted $z_{dominant}$, consists of salient features that are highly predictive of y. The second type, denoted z_{subtle}, represents fine-grained, complementary information that may contribute to generalization but exhibits subtle variations and is more easily overshadowed by noise.

Under this uniform compression strategy, VIB may struggle to retain z_{subtle} while suppressing nuisance variation. Relative to $z_{dominant}$, both z_{subtle} and z_{noise} enter the representation as weak variations, and the global KL penalty does not distinguish between them. As a result, subtle predictive information may be lost, potentially limiting out-of-distribution generalization.

3.2 OVERVIEW OF ADVERISF

To address the limitations of uniform statistical compression in VIB, we propose AdverISF. Instead of compressing the input through a single encoder with a global KL penalty, AdverISF adopts a dual-encoder architecture that explicitly distinguishes task-relevant features z_{task} from nuisance factors z_{noise}. These encoders allocate them into distinct latent representations z_p and z_r. An adversarial mechanism is then introduced to reduce the statistical dependence between the two representations, supporting the separation. Figure 2 illustrates the overall architecture.

3.3 ADVERSARIAL INFORMATION SEPARATION BLOCK

The Adversarial Information Separation Block serves as the fundamental building block of the model. It decomposes input data into dual latent spaces through three key sub-modules. The primary branch captures predictive information, while the residual branch models residual variations and confounding factors. A core challenge in this decomposition is to prevent the two encoders from encoding overlapping information. We address this through an Adversarial Separation Mechanism that reduces statistical dependence between the learned representations. This mechanism acts as a regularizer to encourage separation and avoid information leakage across the two spaces.

3.3.1 PRIMARY BRANCH

The primary branch follows the VIB framework. It distills a compact representation z_p that retains maximal information about the target y while filtering out noise in the input x. This branch consists of two components, a Primary Encoder that maps the input to the latent space and a Primary Predictor that estimates the target variable.

Our objective is to maximize $I(z_p; y) - \beta_p \cdot I(z_p; x)$. Since direct optimization of mutual information terms is computationally intractable, we employ a variational approximation. The Primary Encoder models the posterior distribution $p_\theta(z_p|x)$, and the Primary Predictor approximates the likelihood $q_{\psi_1}(y|z_p)$. This yields the following loss:

$$\mathcal{L}_p = \mathbb{E}_x[\mathbb{E}_{z_p \sim p_\theta(z_p|x)}[-\log q_{\psi_1}(y|z_p)]] + \beta_p \cdot D_{KL}[p_\theta(z_p|x) \parallel r(z_p)], \tag{3}$$

where β_p controls the trade-off between prediction accuracy and information compression.

3.3.2 RESIDUAL BRANCH

The residual branch follows a similar architecture but operates conditionally to capture residual information. It consists of a Residual Encoder and a Joint Predictor. The Residual Encoder extracts latent factors z_r that are complementary to the already extracted z_p. It models the conditional distribution $p_\phi(z_r|x, z_p)$, taking the input x and the primary representation z_p jointly as conditions. The Joint Predictor $q_{\psi_2}(y|z_p, z_r)$ then utilizes both latent codes to predict the target. This design ensures that the combination of z_p and z_r captures the full information content of x required for fitting the training data.

The objective of this branch is to maximize $I(z_r; y \mid z_p) - \beta_r \cdot I(z_r; x \mid z_p)$. The loss function is:

$$\mathcal{L}_r = \mathbb{E}_{x,z_p}\Big[\mathbb{E}_{z_r \sim p_\phi(z_r|x,z_p)}[-\log q_{\psi_2}(y|z_p, z_r)]\Big] + \beta_r \cdot D_{KL}[p_\phi(z_r|x, z_p) \parallel r(z_r)], \tag{4}$$

where β_r controls the compression pressure on z_r and is set to a small value. This minimal compression allows the branch to retain residual patterns, and potentially preserve subtle predictive information that might be discarded by the primary branch.

Role of Joint Prediction Why maximize the mutual information between the residual representation z_r and the target y when such factors are typically task-irrelevant? This design choice is critical for the effectiveness of the adversarial game. Without supervision, the Residual Encoder could degenerate into generating generic Gaussian noise unrelated to the input x. In this trivial case, z_p and z_r would be naturally independent, satisfying the adversarial constraint without enforcing any meaningful purification of z_p. To prevent this degeneration, we employ the Joint Predictor to compel z_r to capture residual information (z_{noise}) that is related to y. The result is a strong adversary that forces the subsequent adversarial separation mechanism to perform a non-trivial separation, thereby ensuring the purity of z_p.

Table 1: Generalization performance of all methods on the two standard regression benchmarks. AEP reports the fraction of training data, Concrete the number of training samples. Best results in **bold**.

Model	AEP 10%	20%	50%	70%	Conc. 30	50	70	100	500
MLP	0.322 ± 0.08	0.444 ± 0.02	0.564 ± 0.02	0.558 ± 0.02	0.207 ± 0.09	0.381 ± 0.04	0.581 ± 0.02	0.649 ± 0.01	0.886 ± 0.00
MLP+WD	0.398 ± 0.02	0.463 ± 0.01	0.572 ± 0.01	0.556 ± 0.03	0.209 ± 0.10	0.384 ± 0.05	0.587 ± 0.02	0.648 ± 0.02	**0.888 ± 0.00**
VIB	0.426 ± 0.05	0.490 ± 0.05	0.578 ± 0.03	**0.604 ± 0.03**	0.270 ± 0.07	0.386 ± 0.05	0.589 ± 0.03	0.641 ± 0.03	0.877 ± 0.01
infoR–LSF	0.346 ± 0.06	0.483 ± 0.05	0.537 ± 0.03	0.571 ± 0.03	0.269 ± 0.05	0.451 ± 0.03	0.585 ± 0.02	0.658 ± 0.02	0.882 ± 0.00
Ours (Joint)	0.430 ± 0.07	**0.501 ± 0.04**	**0.599 ± 0.03**	0.597 ± 0.03	0.496 ± 0.02	0.638 ± 0.01	0.662 ± 0.01	**0.674 ± 0.01**	0.815 ± 0.02
Ours (Two-Stage)	**0.431 ± 0.08**	0.416 ± 0.01	0.423 ± 0.08	0.461 ± 0.12	**0.526 ± 0.02**	**0.643 ± 0.01**	**0.685 ± 0.01**	0.669 ± 0.01	0.829 ± 0.00

3.3.3 Adversarial Separation Mechanism

The Adversarial Separation Mechanism aims to reduce the mutual information $I(z_p; z_r)$. Because direct mutual-information estimation is intractable, we optimize this objective using the distribution-based adversarial strategy described below. This design is inspired by the Domain-Adversarial Neural Networks (DANN) (Ganin et al., 2016). DANN uses adversarial training to align feature distributions across different domains. A discriminator is trained to classify explicit domain labels such as source versus target, while the feature extractor attempts to confuse this classifier.

Unfortunately, our scenario differs fundamentally. We lack explicit labels distinguishing primary from residual information within a single input x. Instead of using explicit labels, we propose a distribution-based adversarial strategy in which the discriminator distinguishes between the joint distribution and the product of the marginal distributions. The adversarial game proceeds as follows:

$$\mathcal{L}_{adv} = \mathbb{E}_{shuffle}[\log D(z_p, z_r')] + \mathbb{E}_{paired}[\log(1 - D(z_p, z_r))]. \tag{5}$$

For a given batch of inputs $X = \{x_1, \ldots, x_n\}$, the encoders generate paired representations $Z_{paired} = \{(z_{j,p}, z_{j,r})\}_{j=1}^{n}$, which serve as samples from the joint distribution $p(z_p, z_r)$. We apply a random permutation to the representations within the batch to construct a shuffled set $Z_{shuffled} = \{(z_{j,p}, z_{\pi(j),r})\}_{j=1}^{n}$, where π denotes a random index permutation. This process breaks the instance-wise correspondence and effectively approximates samples from the product of marginals $p(z_p)p(z_r)$. We provide a theoretical discussion for this mechanism in Appendix A.

3.3.4 Total Optimization Objective

Integrating the objectives from the primary branch, residual branch, and adversarial mechanism, the final training objective is formulated as:

$$\min_{\theta,\phi,\psi} \max_{D} \mathcal{L}_{total} = \mathcal{L}_p + \mathcal{L}_r + \lambda_{adv} \cdot \mathcal{L}_{adv}, \tag{6}$$

where λ_{adv} is a hyperparameter balancing the contributions of primary prediction, residual conditional prediction, and adversarial separation. Parameters θ and ϕ correspond to the Primary Encoder and Residual Encoder, respectively, and $\psi = \{\psi_1, \psi_2\}$ parameterizes the Primary and Joint Predictors.

Further, we use WGAN-GP (Arjovsky et al., 2017; Gulrajani et al., 2017) for adversarial training. The $\hat{z}$ is a random interpolation between paired and shuffled samples, and γ is the penalty coefficient. The discriminator loss with gradient penalty is defined as follows:

$$\mathcal{L}_D = \mathbb{E}_{paired}[D(z_p, z_r)] - \mathbb{E}_{shuffled}[D(z_p, z_r')] + \gamma \mathbb{E}_{\hat{z}}[(\|\nabla_{\hat{z}} D(\hat{z})\|_2 - 1)^2]. \tag{7}$$

3.4 Multi-layer Separation Architecture

As discussed in Section 3.1, predictive components in complex domains vary in their importance, spanning from dominant factors $z_{dominant}$ to subtle ones z_{subtle}. A single Adversarial Information Separation Block may struggle to capture this hierarchy. Driven by the adversarial constraint, it adopts a conservative strategy that prioritizes the most discriminative factors to ensure representational compactness. While effective for separation, this trade-off between compactness and completeness may leave less prominent but critical subtle information within the residual representation.

We address this limitation by extending the single-block design into a multi-layer separation architecture. This framework adopts a progressive extraction strategy by cascading multiple Adversarial

Table 2: Generalization performance ($R^2 \uparrow$) on the **Synthetic Dataset**. Best results in **bold**.

Model	10%	20%	50%	70%
MLP	0.201 ± 0.04	0.230 ± 0.01	0.421 ± 0.01	0.468 ± 0.03
MLP+WD	0.205 ± 0.04	0.232 ± 0.02	**0.434 ± 0.02**	0.470 ± 0.03
VIB	0.159 ± 0.07	0.147 ± 0.04	0.393 ± 0.03	0.478 ± 0.06
infoR–LSF	0.165 ± 0.05	0.147 ± 0.04	0.406 ± 0.03	0.435 ± 0.03
Ours (Joint)	0.179 ± 0.06	**0.254 ± 0.06**	0.423 ± 0.05	0.355 ± 0.08
Ours (Two-Stage)	**0.251 ± 0.07**	0.253 ± 0.09	0.420 ± 0.04	**0.565 ± 0.06**

Table 3: Classification accuracy on **CIFAR-10**. Best results in **bold**.

Model	100	200	300	500	1000
ResNet-18	0.233	0.288	0.323	0.374	0.451
VIB	0.258	0.302	0.345	0.393	0.484
infoR–LSF	0.262	0.320	0.357	**0.429**	**0.542**
Ours (Joint)	**0.280**	0.321	0.361	0.412	0.498
Ours (Two-Stage)	0.262	**0.324**	**0.368**	0.409	0.519

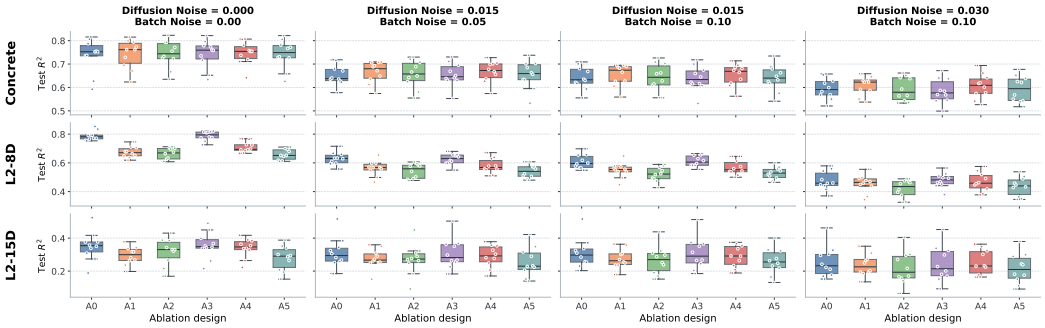

Figure 3: Ablation results under different noise conditions. **A0** denotes the full model; **A1** removes the multi-layer architecture; **A2** removes variational regularization; **A3** removes adversarial separation; **A4** shares the encoder between z_p^1 and z_r^1; and **A5** removes both variational regularization and adversarial separation. Results are test R^2 over 10 runs.

Information Separation Blocks. The residual output from the l-th layer is recycled as input to the $(l+1)$-th layer, enabling the recovery of complementary predictive information not captured by earlier layers. We provide a theoretical discussion for the recovery mechanism in Appendix A.3.

In the first layer, the block takes the raw input x and extracts the dominant predictive information as z_p^1. All remaining information, including z_{subtle} and noise, is pushed into the residual representation z_r^1. In the subsequent layer, the second block operates on z_r^1 to extract the complementary predictive components z_p^2 and isolate the remaining residuals in z_r^2. This process repeats recursively. The comprehensive primary representation is constructed by aggregating the primary codes from all layers, ensuring that both dominant and subtle information are preserved for the final prediction.

4 Experiments

We evaluate AdverISF on benchmark datasets and a real-world materials design task to answer the following questions. (1) Can AdverISF achieve superior generalization performance in data-scarce and out-of-distribution scenarios compared to state-of-the-art baselines? (Sec. 4.2 and Sec. 4.5) (2) Do the proposed adversarial separation mechanism and the multi-layer architecture effectively contribute to the model's performance? (Sec. 4.3 and Sec. 4.4)

4.1 Experimental Setup

We evaluate AdverISF across diverse domains ranging from controlled synthetic environments to real-world applications. The evaluation includes the Concrete Compressive Strength (Yeh, 1998) and Appliances Energy Prediction (AEP) (Candanedo, 2017) datasets from the UCI Machine Learning Repository, CIFAR-10 (Krizhevsky et al., 2009), and a synthetic dataset with explicit ground-truth dominant and subtle factors. To evaluate extreme data scarcity, the training-set size is varied across all datasets. For the ablation and hyperparameter analyses, we use a second synthetic dataset in two input dimensionalities (L2-8D and L2-15D) to control task difficulty, alongside Concrete. Finally, our method is validated on a real-world scientific task in Composite Portland Cement design, which predicts compressive strength from formulations prepared independently in a subsequent experimental

batch, allowing us to assess the value of AdverISF for scientific research under realistic measurement noise and experimental variation.

We compare with MLP, MLP+WD (AdamW with weight decay 10^{-2}), ResNet-18, VIB (Alemi et al., 2017), and infoR-LSF (Xie & Li, 2024), selecting the applicable baselines for each task. We compare two training strategies. Joint training optimizes the entire architecture simultaneously. Two-Stage training optimizes each separation block sequentially. For the comparison experiments, we apply stochastic weight perturbation to the KL divergence terms (Sections 4.2 and 4.5). We report the coefficient of determination (R^2) for regression and accuracy for classification, averaged over 10 runs (mean $\pm$ std). Architectures are detailed in Appendix C.

4.2 Comparison Analysis

We analyze generalization performance across diverse domains to validate the adversarial separation mechanism. On standard benchmarks (Table 1), AdverISF consistently outperforms baselines in extreme data-scarce regimes (e.g., at $N = 30$ on Concrete, the Two-Stage model reaches $R^2 = 0.526$, nearly twice the best baseline at 0.270). Unlike VIB, which suppresses complementary predictive information through uniform compression, our method explicitly preserves critical signals by separating primary and residual representations. We validate this property on the Synthetic dataset (Table 2), where AdverISF achieves the best performance at the most data-scarce ratio and remains competitive across the remaining ratios. Finally, in the high-dimensional CIFAR-10 few-shot task (Table 3), AdverISF outperforms all baselines when only 100 to 300 labeled samples are available. These results confirm that the advantage of AdverISF is concentrated in the extreme data-scarce regime that motivates its design.

4.3 Ablation Analysis

We investigate the main components of AdverISF by comparing the full model (**A0**) with five ablation variants on Concrete and the synthetic L2 datasets under different noise conditions (Fig. 3). The L2 dataset definitions and hyperparameter settings are provided in the Appendix. Compared with the single-layer variant (**A1**), A0 achieves higher median test R^2 in several L2 settings but shows only modest differences on Concrete, suggesting that hierarchical residual extraction is most useful when the task contains complementary predictive structure. The performance gap between the non-adversarial variant (**A3**) and its non-variational counterpart (**A5**) supports the value of variational regularization. The smaller difference between A0 and the non-variational variant (**A2**) suggests that adversarial separation may provide a complementary regularizing effect. Although A0 and A3 achieve broadly comparable median performance, A0 shows lower dispersion under some higher-noise conditions, indicating a potential stability benefit that we further examine through the adversarial-weight analysis. Sharing the encoders for z_p^1 and z_r^1 (**A4**) also reduces performance in some settings, particularly on L2-8D, suggesting that separate encoders may better support representation specialization. Overall, the results show that the contribution of each component depends on the predictive structure and noise conditions of the task.

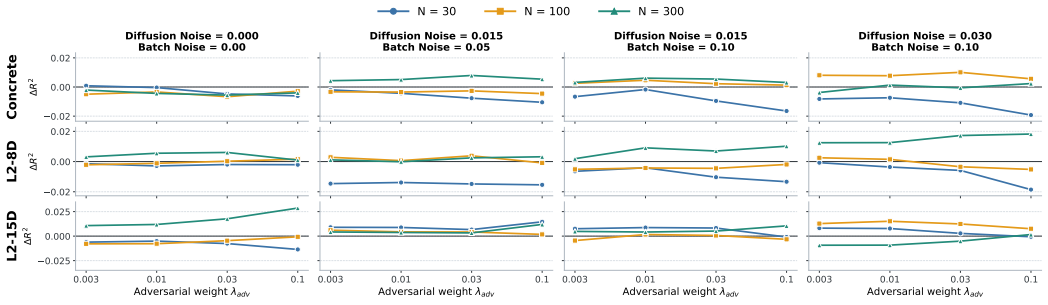

Figure 4: **Effect of the adversarial loss weight λ_{adv} under different noise conditions.** Each panel reports the change in test R^2 relative to the variant without adversarial separation (**A3**) for three training-set sizes N.

Table 4: Performance comparison on the Real-world Task. Best Test result is in bold.

Split	MLP	MLP+WD	VIB	infoR–LSF	Ours (Joint)	Ours (Two-Stage)
Train	1.000 ± 0.00	1.000 ± 0.00	0.999 ± 0.00	0.999 ± 0.00	0.976 ± 0.01	0.976 ± 0.01
Valid	0.061 ± 0.32	0.469 ± 0.16	0.812 ± 0.04	0.830 ± 0.02	0.808 ± 0.03	0.724 ± 0.05
Test	-1.559 ± 0.61	0.017 ± 0.15	0.617 ± 0.08	0.795 ± 0.08	$\mathbf{0.897 \pm 0.04}$	0.882 ± 0.05

4.4 Hyperparameter Analysis

We examine the effect of the adversarial weight λ_{adv} by reporting the change in test R^2 relative to the variant without adversarial separation (**A3**). As shown in Fig. 4, two observations emerge. First, larger gains appear under stronger diffusion and batch noise, while the improvements are smaller in the noise-free setting. This aligns with the role of adversarial separation in reducing the influence of nuisance variation. Second, positive gains occur more frequently on the L2 datasets than on Concrete. Because the L2 targets involve nonlinear interactions among input variables, this indicates that adversarial separation is more valuable when the predictive structure is complex and harder to isolate from residual variation. These observations demonstrate the relevance of AdverISF to real-world scientific tasks involving measurement noise, batch effects, and complex feature interactions. However, increasing λ_{adv} does not always improve performance. An excessively large weight often reduces R^2, especially when data are most scarce ($N = 30$). The reason is that strong adversarial pressure enforces representation independence while discarding predictive information. Overall, a small to moderate λ_{adv} more reliably balances information separation and predictive completeness. Analyses of the latent dimensions and compression weights are provided in Appendix B.

4.5 Real-world Application: Composite Portland Cement Design

We apply AdverISF to compressive strength prediction in Composite Portland Cement mix design, a real materials design task in which a model trained on results from preceding experiments must identify promising formulations for subsequent experiments. The goal is to predict compressive strength based on 8 formulation variables. These include curing age, water-cement ratio, and the contents of clinker, gypsum, slag, fly ash, steel slag, and limestone powder. The training record comprises 60 formulations from preceding experimental batches, randomly split 7:3 for training and validation. The model is evaluated on 9 new formulations selected by predicted compressive strength and prepared in a subsequent independent batch, a batch scale typical for such experiments because compressive-strength measurement must await curing.

As shown in Table 4, the MLP without weight decay fails to generalize and obtains a negative R^2 score of -1.559, indicating that this task is highly prone to overfitting. The VIB model improves the performance to 0.617 through input compression. Building on VIB, infoR-LSF reaches 0.795 by maximizing feature retention. However, this method inadvertently preserves experimental noise alongside useful features, which may limit its extrapolation performance. In contrast, AdverISF outperforms all baselines. Our Joint strategy achieves an R^2 of 0.897 and the Two-Stage strategy reaches 0.882. These results are consistent with adversarial separation reducing the influence of experimental noise. This separation contributes to more robust property predictions for real-world material design.

5 Conclusion

In this work, we propose the paradigm of adversarial information separation, which isolates task-relevant information from confounding noise without explicit separation labels. We introduce AdverISF, a multi-layer framework that reduces statistical dependence between primary and residual representations through a task-guided adversarial separation mechanism. To further refine this separation, it progressively recycles residual representations across layers to recover complementary predictive information that might otherwise be discarded. We evaluate AdverISF on diverse benchmarks and a challenging real-world material design task, demonstrating superior performance over competitors in data-scarce and out-of-distribution scenarios. Ablation analysis further demonstrates the effectiveness of the proposed adversarial separation mechanism and multi-layer architecture under noisy, data-scarce conditions.

AI USE STATEMENT

In this work, we used generative AI tools to formulate mathematical claims, to assist in the writing of proofs, to suggest a structure and propose a title and keywords for this research paper, and for language polishing of the manuscript. We have not used generative AI tools to generate synthetic data sets, to help develop theoretical models or conceptual frameworks, to provide critical ingredients for proving mathematical claims, to propose or refine hypotheses, to design or provide feedback on research methodology or experiments, to implement methods, to clean and reformat data sets, or to interpret results, and assistance with translation and support for qualitative and thematic data analysis are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Jean-Christophe Gagnon-Audet, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. Invariance principle meets information bottleneck for out-of-distribution generalization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 3438–3450. Curran Associates, Inc., 2021.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

Zeyuan Allen-Zhu and Yuanzhi Li. Feature purification: How adversarial training performs robust deep learning. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 977–988, 2022.

Rana Ali Amjad and Bernhard C. Geiger. Learning representations for neural network-based classification using the information bottleneck principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(9):2225–2239, 2020.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 214–223. PMLR, 06–11 Aug 2017.

Philemon Brakel and Yoshua Bengio. Learning independent features with adversarial nets for non-linear ica. *arXiv preprint arXiv:1710.05050*, 2017.

Luis Candanedo. Appliances Energy Prediction. UCI Machine Learning Repository, 2017.

Ricky T. Q. Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

MinGyu Choi and Changhee Lee. Conditional information bottleneck approach for time series imputation. In *The Twelfth International Conference on Learning Representations*, 2024.

Tommy WS Chow and Di Huang. Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information. *IEEE Transactions on Neural networks*, 16(1):213–224, 2005.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Bozheng Dou, Zailiang Zhu, Ekaterina Merkurjev, Lu Ke, Long Chen, Jian Jiang, Yueying Zhu, Jie Liu, Bengong Zhang, and Guo-Wei Wei. Machine learning methods for small data challenges in molecular science. *Chemical Reviews*, 123(13):8736–8780, 2023.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Gokul Gowri, Xiao-Kang Lun, Allon M. Klein, and Peng Yin. Approximating mutual information of high-dimensional variables using learned representations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 132843–132875. Curran Associates, Inc., 2024.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

Kyle Hsu, William Dorrell, James Whittington, Jiajun Wu, and Chelsea Finn. Disentanglement via latent quantization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 45463–45488. Curran Associates, Inc., 2023.

Kyle Hsu, Jubayer Ibn Hamid, Kaylee Burns, Chelsea Finn, and Jiajun Wu. Tripod: three complementary inductive biases for disentangled representation learning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Dou Hu, Lingwei Wei, Wei Zhou, and Songlin Hu. Representation learning with conditional information flow maximization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14088–14103, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Maria CG Juenger, Ruben Snellings, and Susan A Bernal. Supplementary cementitious materials: New sources, characterization, and performance insights. *Cement and Concrete Research*, 122: 257–273, 2019.

Kenji Kawaguchi, Zhun Deng, Xu Ji, and Jiaoyang Huang. How does information bottleneck help deep learning? In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 16049–16096. PMLR, 23–29 Jul 2023.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International conference on machine learning*, pp. 2649–2658. PMLR, 2018.

Junho Kim, Byung-Kwan Lee, and Yong Man Ro. Distilling robust and non-robust features in adversarial examples by information bottleneck. *Advances in Neural Information Processing Systems*, 34:17148–17159, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Huafeng Kuang, Hong Liu, Yongjian Wu, Shin'ichi Satoh, and Rongrong Ji. Improving adversarial robustness via information bottleneck distillation. *Advances in Neural Information Processing Systems*, 36:10796–10813, 2023.

Bo Li, Yifei Shen, Yezhen Wang, Wenzhen Zhu, Dongsheng Li, Kurt Keutzer, and Han Zhao. Invariant information bottleneck for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7399–7407, 2022a.

Zhanzhao Li, Jinyoung Yoon, Rui Zhang, Farshad Rajabipour, Wil V Srubar III, Ismaila Dabo, and Aleksandra Radlińska. Machine learning in concrete science: applications, challenges, and best practices. *npj computational materials*, 8(1):127, 2022b.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6028–6039. PMLR, 13–18 Jul 2020.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Jiashuo Liu, Zheyan Shen, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Massimiliano Mancini, Lorenzo Porzi, Fabio Cermelli, and Barbara Caputo. Discovering latent domains for unsupervised domain adaptation through consistency. In Elisa Ricci, Samuel Rota Bulò, Cees Snoek, Oswald Lanz, Stefano Messelodi, and Nicu Sebe (eds.), *Image Analysis and Processing – ICIAP 2019*, Cham, 2019. Springer International Publishing.

Ziqi Pan, Li Niu, Jianfu Zhang, and Liqing Zhang. Disentangled information bottleneck. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9285–9293, 2021.

Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. In *The 37th Allerton Conference on Communication, Control, and Computing*, 1999.

Deng-Bao Wang and Min-Ling Zhang. Calibration bottleneck: Over-compressed representations are less calibratable. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 52156–52170. PMLR, 21–27 Jul 2024.

Hanchen Wang, Tianfan Fu, Yuning Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

Zhipeng Xie and Yahe Li. Information retention via learning supplemental features. In *International Conference on Learning Representations (ICLR)*, 2024.

Huan Xu and Shie Mannor. Robustness and generalization. *Machine Learning*, 86(3):391–423, 2012.

Pengcheng Xu, Xiaobo Ji, Minjie Li, and Wencong Lu. Small data machine learning in materials science. *npj Computational Materials*, 9(1):42, 2023.

I-Cheng Yeh. Concrete Compressive Strength. UCI Machine Learning Repository, 1998.

Shujian Yu, Luis G Sánchez Giraldo, and José C Príncipe. Information-theoretic methods in deep neural networks: Recent advances and emerging opportunities. In *IJCAI*, pp. 4669–4678, 2021.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7523–7532. PMLR, 09–15 Jun 2019.

Xinqi Zhu, Chang Xu, and Dacheng Tao. Commutative lie group vae for disentanglement learning. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12924–12934. PMLR, 18–24 Jul 2021.

A Theoretical Analysis of Adversarial Information Separation

In this section, we analyze the theoretical guarantees of the proposed adversarial separation mechanism. We demonstrate that enforcing statistical independence between primary representations z_p and residual representations z_r can be theoretically achieved through either Jensen-Shannon Divergence (Vanilla GAN) minimization or Wasserstein distance (WGAN) minimization.

A.1 Independence via Jensen-Shannon Divergence (Vanilla GAN)

Following the standard adversarial framework (Goodfellow et al., 2014), the discriminator D distinguishes between the joint distribution $P_J = p(z_p, z_r)$ and the product of marginals $P_M = p(z_p)p(z_r)$. The minimax objective is formulated as:

$$\min_E \max_D V(D, E) = \mathbb{E}_{\mathbf{x}\sim P_M}[\log(D(x))] + \mathbb{E}_{\mathbf{y}\sim P_J}[\log(1 - D(\mathbf{y}))] \tag{8}$$

Proposition 1. *The global minimum of the standard GAN objective is achieved if and only if the Jensen-Shannon Divergence $D_{JSD}(P_J \| P_M) = 0$.*

Since $D_{JSD}(P\|Q) = 0$ implies $P = Q$ almost everywhere, optimizing this objective theoretically enforces $P_J = P_M$, satisfying the definition of statistical independence.

A.2 Independence via Wasserstein Distance (WGAN)

Alternatively, utilizing the Kantorovich-Rubinstein duality (Arjovsky et al., 2017), the separation can be formulated using a critic D constrained to the set of 1-Lipschitz functions ($\|D\|_L \leq 1$):

$$\min_E \max_{D:\|D\|_L\leq 1} (\mathbb{E}_{\mathbf{y}\sim P_J}[D(\mathbf{y})] - \mathbb{E}_{\mathbf{x}\sim P_M}[D(\mathbf{x})]) \tag{9}$$

Proposition 2. *Optimizing the WGAN objective is equivalent to minimizing the Wasserstein-1 distance $W_1(P_J, P_M)$.*

Since W_1 is a valid metric on the space of probability distributions, convergence $W_1 \to 0$ implies weak convergence $P_J \xrightarrow{d} P_M$, thereby guaranteeing $z_p \perp z_r$.

While both frameworks theoretically ensure independence, we adopt WGAN-GP (Gulrajani et al., 2017) for our implementation. The Wasserstein metric provides consistent, non-vanishing gradients even when high-dimensional latent distributions have disjoint supports, offering superior training stability in our data-scarce domains compared to the saturation-prone JS divergence.

A.3 Hierarchical Feature Recovery via Conditional Mutual Information

While the adversarial mechanism is highly effective at reducing redundancy between representations, the strictness of the independence constraint introduces an inherent limitation regarding predictive completeness. As noted in Section 3.4, the requirement to rigorously satisfy the independence condition $I(z_p; z_r) \to 0$ compels the encoder to prioritize primary predictive signals that are easily separated from residual variation. Consequently, ambiguous or subtle predictive signals are inadvertently suppressed to avoid adversarial penalties.

This phenomenon results in the first separation layer functioning as a stringent filter that yields a compact but potentially incomplete primary representation z_p^1. The residual representation z_r^1 therefore absorbs the residual subtle information as an unavoidable byproduct of the separation process. To address this limitation, we provide an information-theoretic justification for the proposed Multi-layer Separation Architecture and demonstrate its capability to recover these hierarchical predictive components through conditional mutual information maximization.

Let the total task-relevant information in input x be decomposed into primary predictive information F_{primary} and complementary predictive information F_{comp}, such that $I(x; y) = I(F_{\text{primary}}; y) + I(F_{\text{comp}}; y \mid F_{\text{primary}})$.

Step 1: Residual Information in Layer 1. The first separation block minimizes $I(z_p^1; z_r^1)$ while maximizing the task prediction $I(z_p^1; y)$. Simultaneously, the residual branch maximizes the conditional mutual information $I(z_r^1; y \mid z_p^1)$.

Under the conservative separation strategy, the model prioritizes F_{primary} for the primary branch. Consequently, the complementary predictive information is effectively "crowded out" and forced into the residual representation by the joint prediction objective:

$$z_p^1 \approx F_{\text{primary}}, \quad z_r^1 \supseteq F_{\text{comp}} \tag{10}$$

This confirms that z_r^1 is a residual representation rather than pure noise and contains complementary predictive information, *i.e.*, $I(z_r^1; y \mid z_p^1) > 0$.

Step 2: Recovery via Conditional Maximization. The second layer takes z_r^1 as input to extract a new representation z_p^2. Since z_p^1 is fixed, optimizing the second layer is equivalent to maximizing the mutual information from the residual representation space. By the Chain Rule of Mutual Information, the total information extracted is:

$$I(z_p^1, z_p^2; y) = I(z_p^1; y) + \underbrace{I(z_p^2; y \mid z_p^1)}_{\text{Information Gain}} \tag{11}$$

Proposition 3 (Strict Information Gain). *If the first layer is incomplete (i.e., residual predictive information exists: $I(z_r^1; y \mid z_p^1) > 0$) and the second layer has sufficient capacity, then the multi-layer architecture can achieve a strict increase in the total predictive information:*

$$I(z_p^1, z_p^2; y) > I(z_p^1; y) \tag{12}$$

Proof Sketch. Since z_p^2 is derived from z_r^1 to maximize the prediction of y, it targets predictive signals orthogonal to z_p^1 (those assigned to the residual representation in Step 1). Thus, the conditional mutual information term is positive, representing the successful recovery of complementary predictive information F_{comp} that were discarded in the single-layer bottleneck.

B Hyperparameter Sensitivity Analysis

In this section, we provide a detailed sensitivity analysis regarding the latent space configurations (z_p^1, z_r^1, z_p^2), the KL divergence coefficients (β_p^1,β_r^1, β_p^2), and the adversarial loss weight (λ_{adv}). The goal is to empirically justify the architectural choices and hyperparameter settings used in our main experiments. We analyze hyperparameter sensitivity on the synthetic dataset (Appendix C.1.1) under varying noise conditions. "Ratio 0.3" and "Ratio 0.7" denote training with 30% and 70% of the dataset, respectively.

B.1 Impact of Latent Space Dimensionality

To understand how the capacity of different latent components affects model performance, we conducted a grid search over the dimensions of the first-stage variables (z_p^1, z_r^1) and the second-stage primary representation (z_p^2), as visualized in the heatmaps of Figures 5 through 8.

For the first-stage primary representation z_p^1, we observed that compact dimensions (e.g., 1 or 4) yield superior performance. Surprisingly, increasing the dimension to 8 resulted in a general degradation of performance. We hypothesize that an overly large z_p^1 induces path dependence, where the model tends to encode all information—including noise and nuisances—into this single variable, rendering the subsequent hierarchical structure redundant. In this scenario, our model effectively degenerates into a standard Variational Information Bottleneck (VIB) baseline.

Conversely, for the residual representation z_r^1, insufficient capacity leads to suboptimal results. A small dimension forces the model to discard complex nuisance variations or leak them into the primary representation, thereby interfering with effective feature extraction. Therefore, the design of the first stage requires a trade-off: a compact primary representation to force abstraction, paired with sufficient residual-branch capacity to absorb task-irrelevant variations.

Regarding the second-stage primary representation z_p^2, the model exhibits greater robustness. As the dimension varies from 1 to 8, performance fluctuations remain relatively mild, with intermediate values (e.g., 4) consistently achieving high stability (see Figures 5 and 7). Consequently, our final configuration adopts a compact z_p^1 to prevent model degeneration, while allocating adequate dimensions to z_r^1 to ensure effective separation of the residual representation.

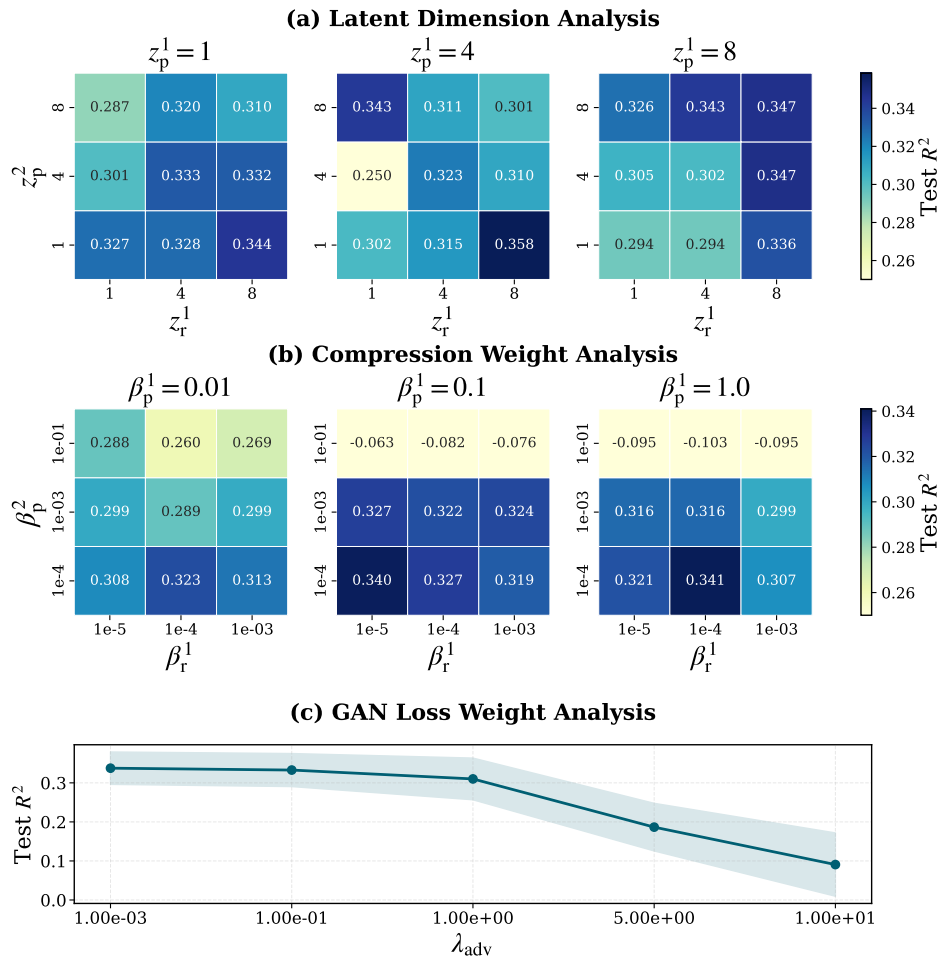

Figure 5: Hyperparameter sensitivity analysis on **Ratio 0.3** using **Joint Training**. The heatmap and curves illustrate performance stability across latent dimensions and weights.

B.2 Interplay of Compression Weights

We further investigated the interaction effects of the compression weights (KL divergence weights): β_p^1, β_r^1, and β_p^2. β_p^1 acts as the main compression control for the first-stage primary representation. Our results indicate that a smaller weight (e.g., $\beta_p^1 \approx 0.01$) provides the most robust performance. As β_p^1 increases to 0.1 or 1.0, the model's tolerance for other hyperparameters decreases significantly. In high-compression regimes, the latent variables are forced to overly approximate the prior, leading to the loss of task-critical information.

A notable finding is the model's extreme sensitivity to the second-stage primary-representation weight, β_p^2. Setting β_p^2 to a relatively large value (e.g., 10^{-1}) causes a catastrophic drop in performance in most configurations of this analysis, resulting in negative R^2 scores (visible as the light-colored regions in Figures 6 and 8). This suggests that excessive regularization at the second stage restricts the model's ability to model complementary predictive information from the residual representation, thereby preventing its recovery. Conversely, a smaller weight (10^{-3} or 10^{-4}) provides the necessary flexibility for stable modeling.

The influence of β_r^1 is relatively mild. Given reasonable settings for the primary-representation weights, β_r^1 maintains good performance across the range of 10^{-5} to 10^{-3}, with 10^{-4} often representing a local optimum. For the model scale analyzed in this parameter analysis, we recommend a light

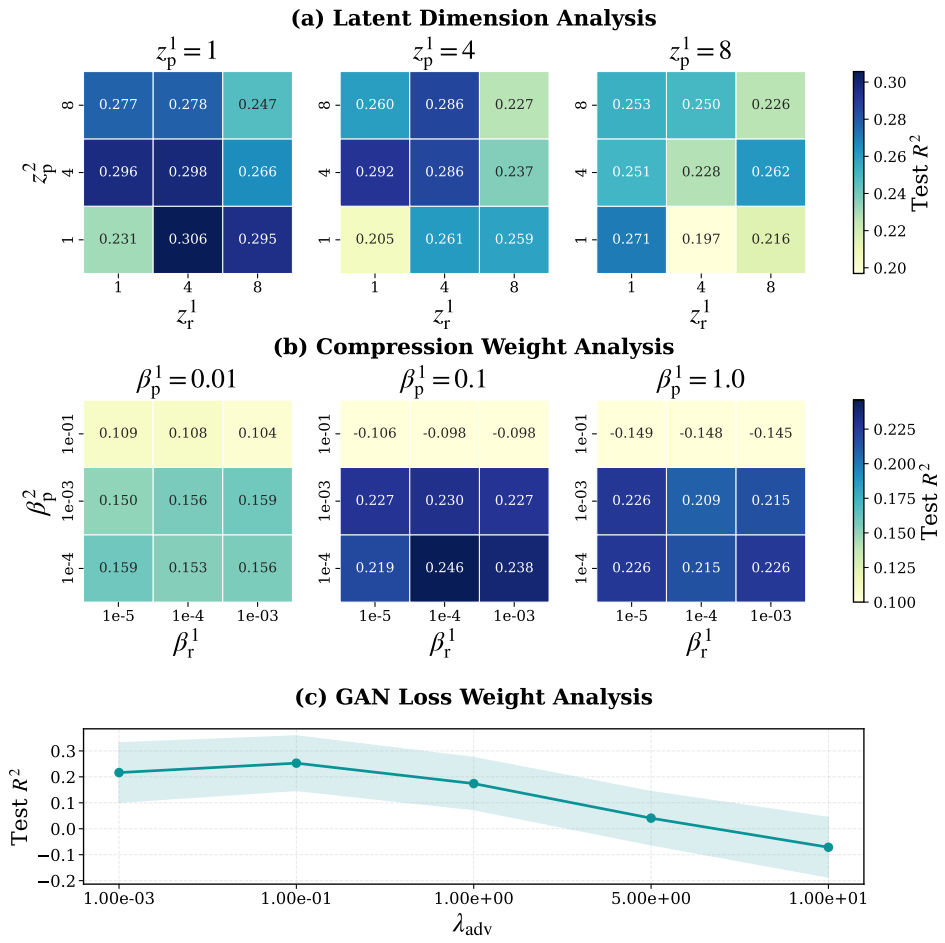

Figure 6: Hyperparameter sensitivity analysis on **Ratio 0.3** using **Two-Stage Training**. Comparison with Joint Training reveals different sensitivity patterns in the latent space.

compression strategy to prevent posterior collapse and ensure effective information flow, maintaining a minimal β_p^2 ($\leq 10^{-3}$) combined with a moderately small β_p^1 ($0.01 - 0.1$).

B.3 Adversarial Training Weight

Finally, we evaluated the impact of the adversarial loss weight, λ_{adv}. We tracked the test set R^2 across different data ratios and training strategies.

The results reveal that adversarial training requires careful balancing. As shown in the line plots of Figure 5 (Ratio 0.3) and Figure 7 (Ratio 0.7), when λ_{adv} is kept within a lower range (e.g., 10^{-3} to 10^{-1}), the model demonstrates optimal stability and accuracy within this sensitivity analysis. At $\lambda_{adv} = 10.0$, the model effectively fails. This phenomenon indicates that excessive adversarial constraints disrupt the structural integrity of the latent space. Therefore, the appropriate value of λ_{adv} depends on the model configuration and the target task, and should be tuned jointly with the other regularization hyperparameters rather than prescribed by a universal bound.

C Implementation Details

In this section, we provide the network architectures and detailed hyperparameter configurations used across our experiments.

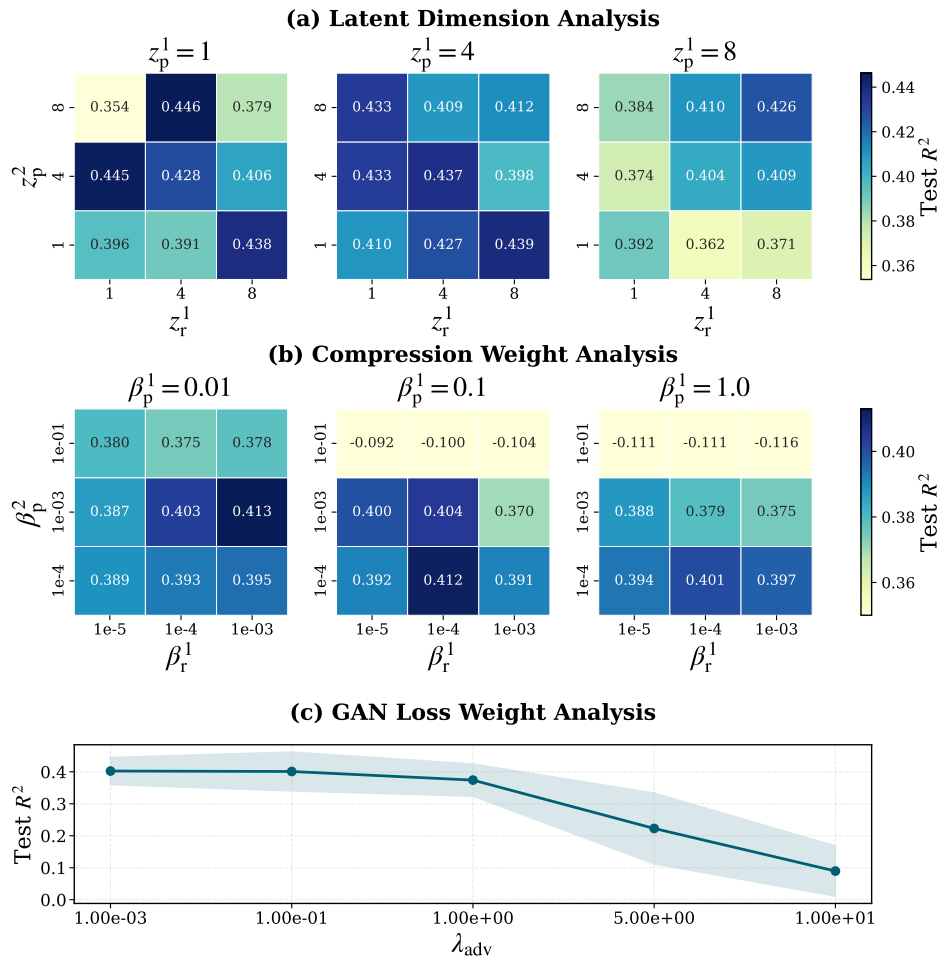

Figure 7: Hyperparameter sensitivity analysis on **Ratio 0.7** using **Joint Training**. With more training data, the model generally shows improved robustness in the sweet spot regions.

C.1 Dataset Construction

C.1.1 Synthetic Dataset

To validate the model's ability to extract hierarchical information from high-dimensional data, we construct a synthetic dataset generated from a common latent source. We employ a non-linear projection mechanism to derive task-relevant latent factors. The data generation pipeline consists of three stages: source sampling, latent feature projection, and target/observation generation.

Source Sampling. We begin by sampling a high-dimensional source vector $s \in \mathbb{R}^{D_{total}}$ from a standard normal distribution $\mathcal{N}(0, I)$. This vector contains all the primitive stochastic information of the system. We set $D_{total} = 13$.

Latent Feature Projection. To simulate the complex extraction of meaningful features from raw data, we map the source vector s to a semantic latent space using a randomly initialized Multi-Layer Perceptron (MLP), referred to as the *Encoder*:

$$z_{base} = \text{MLP}_{enc}(s) \tag{13}$$

where $z_{base} \in \mathbb{R}^{d_1+d_2}$. We then partition this feature vector into two distinct components:

1. **Ground-truth Dominant Factors** ($z_{dominant}$): The first $d_1 = 3$ dimensions of z_{base}, representing high-level core features.

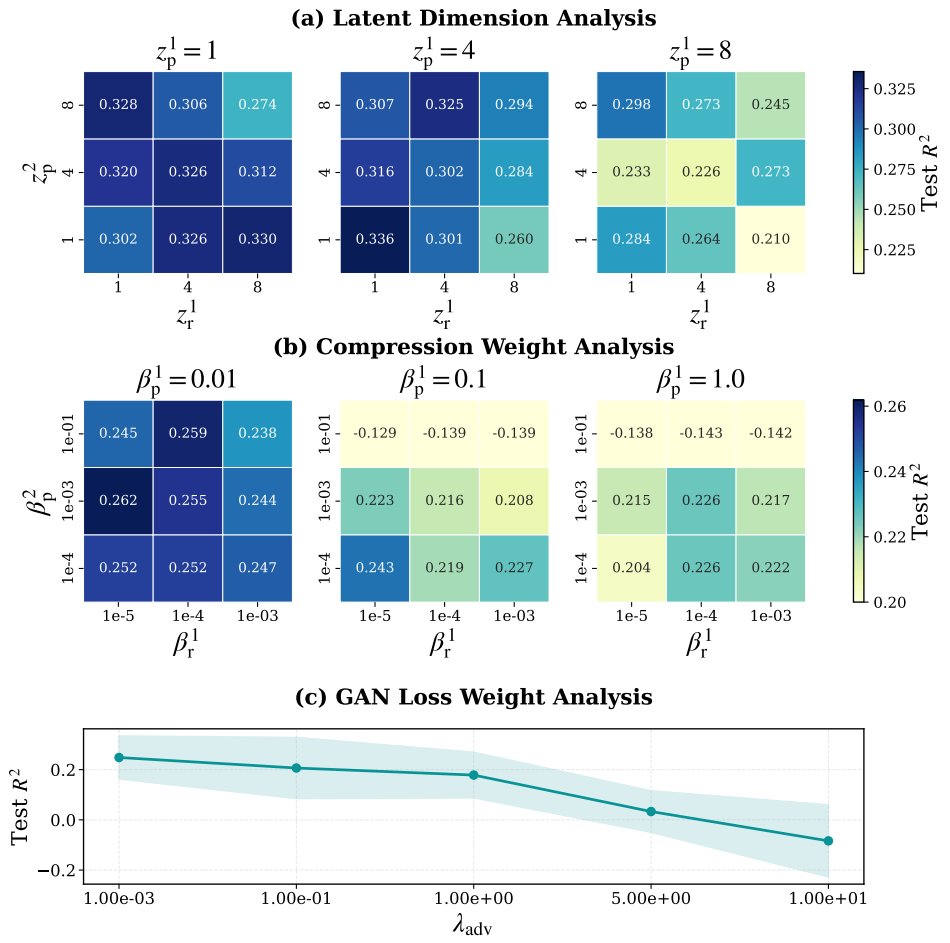

Figure 8: Hyperparameter sensitivity analysis on **Ratio 0.7** using **Two-Stage Training**. The wider confidence intervals at high λ_{adv} indicate potential instability.

2. **Ground-truth Subtle Factors** (z_{subtle}): The subsequent $d_2 = 5$ dimensions, representing fine-grained details.

This projection ensures that the latent factors are non-linear transformations of the original source, creating a more challenging manifold learning task.

Hierarchical Target Generation. The regression target y is derived from these projected factors using two separate random MLPs, f_{dom} and f_{sub}:

$$y = f_{dom}(z_{dominant}) + \gamma \cdot f_{sub}(z_{subtle}) + 0.15 \cdot \epsilon_y \tag{14}$$

We set $\gamma = 0.2$ to scale down the contribution of subtle factors, making them harder to detect. Here, ϵ_y denotes synthetic target noise and is distinct from the learned residual representation z_r.

Observation Entanglement. The model observes a transformed version of the raw source vector s. The input observation x is generated via a linear mixing transformation of the source:

$$x = \mathbf{W}_{trans} \cdot s + \mathbf{b}_{trans} \tag{15}$$

where $\mathbf{W}_{trans} \in \mathbb{R}^{D_{total} \times D_{total}}$ is a random matrix. This implies that x contains all the information present in s, but the semantic features $z_{dominant}$ and z_{subtle} are implicit and must be recovered by inverting the relationship between x and y.

Table 5: Detailed training configurations for baseline models. d_z denotes the latent dimension for VIB. Values for infoR-LSF on CIFAR-10 are aligned with its official implementation.

Dataset	Method	Batch Size	Epochs	LR	Special Configs
AEP	MLP	128	4000	3×10^{-4}	-
	VIB	128	4000	3×10^{-4}	$d_z = 6, \beta = 0.2$
	infoR-LSF	128	4000	3×10^{-4}	-
Synthetic	MLP	20	2000	3×10^{-4}	-
	VIB	20	2000	3×10^{-4}	$d_z = 4, \beta = 5 \times 10^{-4}$
	infoR-LSF	20	2000	3×10^{-4}	-
Concrete	MLP	20	4000	3×10^{-4}	-
	VIB	20	4000	3×10^{-4}	$d_z = 4, \beta = 0.05$
	infoR-LSF	20	4000	3×10^{-4}	-
CIFAR-10	ResNet-18	256	200	5×10^{-4}	-
	VIB	256	200	5×10^{-4}	$d_z = 64, \beta = 0.1$
	infoR-LSF	256	200	0.1	SGD†, WD=5×10^{-4}, StepLR

† For CIFAR-10, infoR-LSF uses SGD with momentum 0.9 and a step decay scheduler.

Table 6: Hyperparameter configurations for our method. We report settings for both Joint and Two-Stage training. The latent space includes primary representations z_p^1, z_p^2, and the residual representation z_r^1. KL weights for primary representations follow a normal distribution $\beta \sim \mathcal{N}(\mu, \sigma^2)$. Fixed weights are reported as single values.

Dataset	Method	Batch Size	Epochs	LR	λ_{adv}	Latent Dims		KL Weights (μ/σ)		
						z_p^1, z_p^2	z_r^1	β_p^1	β_p^2	β_r^1
AEP	Joint	128	16000	2×10^{-4}	1.0	3, 3	5	0.60/0.05	0.15/0.02	8×10^{-4}
	Two-Stage	128	8000×2	$3 \rightarrow 2 \times 10^{-4}$	1.0	3, 3	5	0.27/0.05	0.15/0.02	8×10^{-4}
Synthetic	Joint	20	4000	2×10^{-4}	1.0	2, 2	5	0.25/0.001	$4 \times 10^{-4}/1 \times 10^{-5}$	8×10^{-5}
	Two-Stage	20	3000 + 2000	$3 \rightarrow 2 \times 10^{-4}$	1.3	2, 2	5	0.06/0.001	$4 \times 10^{-4}/1 \times 10^{-5}$	8×10^{-5}
Concrete	Joint	20	4000	2×10^{-4}	1.0	2, 2	5	0.25/0.01	0.05/0.001	8×10^{-5}
	Two-Stage	20	4000×2	$3 \rightarrow 2 \times 10^{-4}$	1.3	2, 2	5	0.06/0.005	$0.005/5 \times 10^{-4}$	8×10^{-5}
CIFAR-10	Joint	256	200	3×10^{-4}	1.0	32, 32	250	0.80/0.03	0.15/0.01	8×10^{-4}
	Two-Stage	256	200×2	3×10^{-4}	1.0	32, 32	250	0.60/0.03	0.10/0.005	8×10^{-4}

C.1.2 L2 DATASET

For the ablation analysis in Section 4.3 (Figure 3) and the hyperparameter analysis in Section 4.4 (Figure 4), we construct a separate synthetic dataset with controlled nonlinear structure, L2-8D and L2-15D. Each sample's input $x_i \sim U(-1, 1)$ for $i = 1, \ldots, d$ where $d \in \{8, 15\}$. The clean target is

$$y_{\text{clean}} = \sum_{i=1}^{d} x_i^2 + \sum_{i=1}^{d} x_i x_{\pi(i)}, \tag{16}$$

where $\pi(i)$ is a fixed random permutation with $\pi(i) \neq i$. Targets are standardized before noise injection.

We inject three types of noise. (1) **Diffusion noise** on inputs, $\tilde{x} = \sqrt{1 - \beta_{\text{diff}}} x + \sqrt{\beta_{\text{diff}}} \epsilon_x$ where $\epsilon_x \sim \mathcal{N}(0, I)$. (2) **Batch noise**, samples are grouped into 10 batches, each sharing $b_g \sim \mathcal{N}(0, \sigma^2_{\text{batch}})$. (3) **Sample noise**, $\epsilon_i \sim \mathcal{N}(0, 0.01^2)$ per sample. The final target is $\tilde{y}_i = y_{\text{std},i} + b_{g(i)} + \epsilon_i$. Figures 3 and 4 evaluate four configurations with $(\beta_{\text{diff}}, \sigma_{\text{batch}}) \in \{(0, 0), (0.015, 0.05), (0.015, 0.10), (0.030, 0.10)\}$.

Each dataset has 500 samples split 70/15/15 for training pool, validation, and test. Training uses $N \in \{30, 100, 300\}$ subsamples. Each setting is repeated 10 times.

C.2 Network Architectures

Feature Backbones. For tabular datasets (AEP, Concrete, Synthetic, Cement), we employ Multi-Layer Perceptrons (MLPs) with LeakyReLU activations. For the CIFAR-10 image dataset, we adopt a ResNet-18 backbone with the final fully-connected layer removed to extract latent features.

Model-Specific Configurations.

- **Ours.** For tabular tasks, the encoders and predictors are MLPs with 2 hidden layers of sizes $[100, 100]$. The discriminator is an MLP with 2 hidden layers of sizes $[50, 50]$, taking the concatenation of primary and residual representations as input. To ensure training stability, we implement the adversarial component using the Wasserstein GAN with Gradient Penalty (WGAN-GP) framework, where the discriminator (critic) is updated $n_{critic} = 2$ times per iteration of the encoders. For the ablation studies (Figure 3), we use wider networks with hidden dimensions [256, 256] and Batch Normalization. The latent dimensions are fixed as $\dim(z_p^1) = 4$, $\dim(z_r^1) = d$ (input dimension), and $\dim(z_p^2) = 6$. For CIFAR-10, the encoder utilizes the ResNet-18 backbone, while the remaining components follow the MLP-based design.

  To enhance training stability and representation diversity, we apply a stochastic weight perturbation strategy to the KL divergence terms. Instead of using fixed coefficients, we sample the penalty weight β for each KL term from a normal distribution $\mathcal{N}(\mu, \sigma^2)$ for every training batch. This randomized scaling forces the model to adapt to varying regularization strengths. This process encourages the discovery of more robust and disentangled features. The specific μ and σ values are determined based on the hyperparameter configurations for each dataset.

- **MLP.** We employ an MLP with 3 hidden layers of sizes $[400, 400, 400]$ for these baselines. Since these models operate without a latent bottleneck, we adopt a shallower architecture. To ensure a fair comparison, we increase the hidden dimension to 400 to maintain a total parameter count consistent with the proposed method.

- **ResNet-18.** We adopt the ResNet-18 architecture as implemented in the infoR-LSF repository.

- **VIB.** The encoder and predictors are MLPs with 3 hidden layers with hidden dimensions of $[200, 200, 200]$. The output dimension of encoder equals the sum of the dimensions of the two MLPs for z_p^1 and z_p^2. This component maps the input to a d_z-dimensional latent space. For CIFAR-10, the encoder utilizes the ResNet-18 backbone, while the remaining components follow the MLP-based design.

- **infoR-LSF.** We strictly follow the default architecture and hyperparameter settings from the official repository.

C.3 Hyperparameter Settings

Baseline Configurations. Table 5 summarizes the training settings for each baseline. Unless otherwise noted, all models are optimized using the Adam optimizer. For infoR-LSF on CIFAR-10, we adopt the original SGD configuration to maintain consistency with its reported performance.

Ours Configurations. Table 6 details the hyperparameters for our proposed method, including the specific distributions used for randomized KL weight sampling.

C.4 Ablation Study Setup

To verify the effectiveness of our proposed mechanisms, we design a comprehensive ablation study on L2 synthetic datasets (Appendix C.1.2) and Concrete.

Model Variants We systematically disable key components to isolate their contributions.

- **A0 (Full Model)** The complete AdverISF with two-layer hierarchy, variational inference, and adversarial separation.

- **A1 (w/o Multi-layer)** Single-layer model. The second separation block is disabled by freezing its parameters and generating predictions solely from the first level.
- **A2 (w/o Variational)** Removes variational bottleneck. The reparameterization trick is disabled and all KL divergence terms are omitted.
- **A3 (w/o Adversarial)** Removes the discriminator and sets $\lambda_{\text{adv}} = 0$. The model is trained purely on prediction and variational objectives.
- **A4 (Shared Encoder)** Tests whether separate encoders for z_p^1 and z_r^1 are necessary. Uses a single shared encoder backbone that outputs a concatenated representation of width $\dim(z_p^1) + \dim(z_r^1)$, which is then split into z_p^1 and z_r^1, rather than two independent encoder networks.
- **A5 (w/o Variational and Adversarial)** Removes both variational regularization (all β terms) and adversarial separation ($\lambda_{\text{adv}} = 0$), reducing the model to a standard dual-encoder architecture.

Hyperparameters The latent dimensions are $\dim(z_p^1) = 4$, $\dim(z_r^1) = d$ (8 for Concrete and L2-8D, 15 for L2-15D), and $\dim(z_p^2) = 6$. The full model (A0) uses KL weights $(\beta_p^1, \beta_r^1, \beta_p^2) = (0.1, 0.0002, 0.0025)$ and adversarial weight $\lambda_{\text{adv}} = 0.03$ for Figure 3. The prediction loss weights three branches as 1.0, 0.5, and 1.0 respectively. Both encoder and discriminator learning rates are 3×10^{-4}. The discriminator is updated 4 times per encoder update, with gradient penalty coefficient $\gamma = 10$.

Table 7: Hyperparameter settings for the Real-world Application.

Category	**Value / Setting**
Training & Architecture	
Optimizer	Adam (Learning Rate: 5×10^{-4}, Batch Size: 20)
Training Schedule	15000 Epochs (Patience: 500)
MLP Hidden Dim	30 (Reduced from 400)
Latent Dimensions	$z_p^1 = 2$, $z_p^2 = 2$, $z_r^1 = 5$
Method-Specific Coefficients	
VIB	$d_z = 4$, $\beta = 0.2$
Ours (Joint)	$\lambda_{adv} = 1.0$ $\beta_p^1 \sim \mathcal{N}(\mu = 1.5, \sigma = 0.05)$, $\beta_p^2 \sim \mathcal{N}(\mu = 0.2, \sigma = 0.01)$ $\beta_r^1 = 8 \times 10^{-5}$
Ours (Two-Stage)	$\lambda_{adv} = 1.3$ $\beta_p^1 \sim \mathcal{N}(\mu = 0.9, \sigma = 0.1)$, $\beta_p^2 \sim \mathcal{N}(\mu = 0.1, \sigma = 0.01)$ $\beta_r^1 = 8 \times 10^{-5}$

C.5 Hyperparameter Analysis Setup

We conduct a comprehensive ablation study on the synthetic dataset to evaluate the sensitivity of our model to structural and regularization hyperparameters. Experiments are performed under two data regimes: using 30% and 70% of the data for training.

Training Strategies. We compare two optimization schedules:

1. **Joint Training:** A standard end-to-end approach where all latent levels are optimized simultaneously for 5,000 epochs.
2. **Two-Stage Training:** A stepwise approach designed to stabilize hierarchical learning. Phase 1 (0–2k epochs) optimizes z_p^1 and z_r^1 while freezing z_p^2. Phase 2 (2k–5k epochs) freezes z_p^1 and z_r^1 to focus exclusively on refining z_p^2.

Hyperparameter Configurations. We analyze performance variations across three axes. For robustness, each configuration is repeated over 10 independent trials.

- **Latent Dimensions:** We vary the size of each latent level $z_p^1, z_r^1, z_p^2 \in \{1, 4, 8\}$ to test representation capacity.
- **Regularization Weights:** We assess the impact of the variational bottleneck and adversarial constraints by varying the KL divergence weights ($\beta_p^1, \beta_r^1, \beta_p^2$ ranging from 10^{-5} to 1.0) and the adversarial loss weight (λ_{adv}) (ranging from 0.001 to 10.0) with $n_{critic} = 6$.

C.6 Real-world Application Setup (Composite Portland Cement)

Dataset and Protocol. The dataset comprises 8 input variables: Curing Age, Water-Cement Ratio, and the mass percentages of six components. To strictly enforce a "future prediction" setting, we utilized 60 historical samples for training and validation (randomly split 7:3) and reserved 9 samples synthesized in a separate, subsequent batch of experiments for testing. Inputs were normalized based on training set statistics to prevent leakage. Due to the extremely limited sample size, we reduced the hidden layer dimension of all MLPs from 400 to 30 to prevent overfitting.

Implementation Details. We employed the Adam optimizer with early stopping. Detailed configurations for the training process, network dimensions, and method-specific hyperparameters are summarized in Table 7.