CONDE: DECOMPOSING REPRESENTATIONS INTO MANIPULABLE SEMANTIC COMPONENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Disentangling semantic factors within learned representations is critical for targeted analysis, manipulation, and improved downstream performance. However, a general disentanglement formulation applicable post-hoc to frozen representations, for multiple semantic factors, and diverse real-world data domains remains underexplored. In this work, we introduce the Contrastive Decomposer (ConDe), a weakly supervised framework designed for this general-case setting. ConDe decomposes an embedding into any number of components whose element-wise sum reconstructs the representation. Because all components inhabit the same latent space, they can be inspected independently and directly recombined, allowing for direct latent manipulation of semantic factors. ConDe supports strict and soft pairings and can be trained end-to-end or applied post hoc to frozen encoders by leveraging invertible networks. We further propose an evaluation protocol for supervised vector-wise disentanglement that assesses both the learned components and their downstream effects, and introduce SynBeeps, a controlled audio benchmark for testing partial disentanglement under complex, perturbed factor relationships. Experiments across vision, audio, and geological imaging show that ConDe learns informative semantic components that support direct manipulation and improved generalization under domain shifts.

1 INTRODUCTION

Modern machine learning applications increasingly rely on learned representations rather than manually designed features. In deep learning, these representations are typically encoded as embeddings in a high-dimensional latent space, whose geometry reflects high-level information about the input. However, standard training objectives primarily optimize predictive or reconstructive performance, without constraining how semantic information is organized within the latent space. Consequently, attributes such as color, shape, texture, or object identity may be encoded jointly across the same dimensions or subspaces. This phenomenon is referred to as entanglement: different semantic factors of variation are represented in a coupled manner, such that changes to one factor cannot be easily isolated from changes to others, and the individual contribution of a given dimension or subspace to any specific semantic property becomes difficult to identify.

This coupling may serve as an obstacle both to interpreting learned representations and to applications that require directly operating on specific semantic factors. For example, controlled generation and style transfer require changing a target attribute while preserving unrelated content, domain-invariant tasks require suppressing environment-specific information while retaining task-relevant features, and representation analysis requires attributing observed behavior to individual, identifiable factors. Disentangled representation learning (DRL) (Bengio et al., 2013; Higgins et al., 2018) addresses these objectives by encouraging distinct generative or semantic factors to occupy separable components of the representation. Disentangled representations have been used for style transfer, domain invariance, controlled generation, and others (Karras et al., 2019; Klys et al., 2018; Norman & Meir, 2026; Bousmalis et al., 2016; Bereket & Karaletsos, 2023; Esser et al., 2020).

DRL methods differ in several key assumptions. Early formulations largely adopted a dimension-wise view, aligning each factor with a single latent coordinate (Higgins et al., 2017; Kim & Mnih, 2018). Although suitable for simple factors, this axis-aligned assumption can be restrictive for concepts whose variation is intrinsically multidimensional. Vector-wise approaches instead use a mul-

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.

tidimensional component, offering greater capacity for complex semantic attributes (Bouchacourt et al., 2018; Fumero et al., 2021). Methods also differ in how factors are specified: unsupervised approaches recover them from data alone, while supervised and weakly supervised approaches use labels or relational information to target predetermined attributes, which is crucial since disentangled representations are otherwise unidentifiable (Locatello et al., 2019a). Disentanglement methods also differ in scope. Whereas exhaustive formulations model all variation, partial decomposition isolates only selected factors and preserves the remainder in a residual.

While many methods employ an end-to-end procedure in which the encoder is directly trained to structure the latent space, some approaches instead adopt post-hoc settings to improve the interpretability and manipulation of pretrained representations. Invertible Interpretation Networks (IINs) (Esser et al., 2020) and Factors Decomposer-Entangler Networks (FDENs) (Yoon et al., 2021), for example, reorganize frozen representations into semantically meaningful components while preserving the information needed to recover the original embedding. These methods demonstrate that semantic structure can be recovered directly from a pretrained representation, but their formulations commit each factor to a fixed dimensionality, which can provide either insufficient or unnecessary representational capacity. Moreover, existing objectives provide little control over how much of a semantic concept is assigned to a component, making them less suitable for settings where only selected factors should be isolated while all remaining information must be preserved.

To address this setting, we introduce the CONtrastive DEcomposer (ConDe), a framework for factor-guided decomposition of learned representations. ConDe leverages relational supervision to map an input representation into any number of vector-valued semantic components and a residual component, which are combined through addition. In the post-hoc setting, an invertible neural network (INN) first transforms a pretrained representation into a coordinate system more amenable to additive decomposition, while preserving the information required to recover the original embedding. The additive formulation avoids partitioning the latent coordinates into fixed factor-specific blocks, allowing each factor component to use the shared ambient representation space, and each component to distribute its information according to the complexity of the corresponding semantic attribute. Because the components remain in a common latent space and can be inspected, removed, or exchanged directly, the decomposition also provides a more interpretable interface for analyzing and manipulating pretrained representations than approaches that recover them through a learned entangling network. ConDe further supports an end-to-end training regime capable of directly learning task-appropriate disentangled latent spaces for complex tasks. Furthermore, its formulation can be relaxed to function under soft pairing, abandoning the assumption that positive examples must share identical semantic information.

Our contributions are (i) ConDe, a weakly supervised framework that decomposes frozen representations into any number of additive, vector-valued components and a residual, with an invertible transformation for post-hoc use and an end-to-end variant; (ii) an evaluation protocol for supervised, vector-wise, partial decomposition, together with SynBeeps, a benchmark combining specified factors and unmodeled perturbations; and (iii) evaluations across latent transfer, domain generalization, image style transfer, music timbre transfer, and seismic interpretation.

2 Related Work

Disentangled representation learning seeks to organize latent representations such that distinct factors of variation are encoded in separable components (Bengio et al., 2013). Early approaches predominantly adopted an unsupervised, dimension-wise formulation, encouraging each factor to align with an individual latent coordinate. Methods such as β-VAE, FactorVAE, and β-TCVAE promote this structure by constraining latent capacity or reducing statistical dependence among dimensions (Higgins et al., 2017; Kim & Mnih, 2018; Chen et al., 2018). Although effective in controlled benchmarks, this formulation assumes that meaningful factors can be represented by scalar coordinates, which are often insufficient for complex features (Higgins et al., 2018; Wang et al., 2024). Moreover, unsupervised disentanglement is not identifiable without additional inductive biases (Locatello et al., 2019a), motivating the use of supervision to select the intended factorization.

Supervised and Vector-wise Disentanglement Methods differ in both the supervision they require and the representational capacity given to each factor. Direct approaches use factor labels to associate predefined semantics with individual latent coordinates, often through auxiliary prediction

objectives (Locatello et al., 2019b). Other methods condition the latent distribution on auxiliary variables or synthetic controls (Khemakhem et al., 2020; Deng et al., 2020), while CSVAE extends direct supervision to multidimensional subspaces (Klys et al., 2018).

Relational approaches instead specify which factors are shared across paired or grouped observations without requiring exhaustive labels. ML-VAE aggregates observations from the same group into a shared representation (Bouchacourt et al., 2018), pairwise methods align components according to known similarities (Chen & Batmanghelich, 2020), product-manifold projection identifies factor-specific submanifolds from paired samples (Fumero et al., 2021), and, when the shared factor is unknown, low-divergence coordinates are treated as shared and averaged across the pair (Locatello et al., 2020). Such relational supervision is useful when factors are easier to define through equivalence than through exhaustive annotation. Vector-wise relational formulations appear in many practical applications such as domain separation, content–style factorization in images and text, biological perturbation modeling, and audio transfer (Bousmalis et al., 2016; Kotovenko et al., 2019; John et al., 2019; Bereket & Karaletsos, 2023; Cífka et al., 2021).

Post-hoc Disentanglement methods operate directly on frozen pretrained representations, using learned transformations to reorganize the latent space. The Factors Decomposer–Entangler Network (FDEN) decomposes an embedding into fixed-dimensional vectors and reconstructs it from their concatenation (Yoon et al., 2021). Supervised predictors associate each component with a predefined attribute, while unmodeled information is assigned to a residual component. FDEN can therefore be interpreted as a vector-wise method supporting partial decomposition through direct supervision. However, as we show in Section 4, its classification objective only requires each component to be predictive of its assigned attribute and may be satisfied by a few discriminative cues, leaving other information associated with the same factor in the residual. Moreover, because the learned entangler only approximates the original embedding, the decomposition may discard or distort information, thereby reducing both downstream reconstruction quality and interpretability.

Invertible Interpretation Networks (IINs) avoid this reconstruction bottleneck by learning a bijective transformation from the original embedding to a partitioned latent space (Esser et al., 2020). Positive pairs are encouraged to correlate in the component assigned to their shared factor and remain independent elsewhere. Factor dimensionalities are estimated beforehand from covariance patterns, with unused dimensions forming a residual. Although invertibility preserves the complete representation, it does not guarantee that each partition captures all information about its factor, and an inaccurate dimensionality estimate may restrict its capacity. In this sense, FDEN and IIN are therefore the closest post-hoc baselines to ConDe. In contrast, ConDe combines invertible reparameterization with additive factor components, relational supervision, and a residual, while supporting multiple factors, partial decomposition, and both strict and soft pairing.

Evaluation of Disentangled representations remains a central challenge for the field. The Mutual Information Gap (MIG) computes the mutual information between latent dimensions and generative factors, evaluating whether each factor is concentrated in a single dimension (Chen et al., 2018). DCI trains supervised factor probes on the latent representation and uses their feature-importance matrix to quantify disentanglement, completeness, and informativeness (Eastwood & Williams, 2018). However, both metrics reflect the dimension-wise assumptions of early disentanglement frameworks. Recent evaluation protocols have adapted MIG and DCI to vector-valued representations (Barami et al., 2026; Fumero et al., 2021; Sha & Lukasiewicz, 2024).

3 METHODOLOGY

ConDe is a framework for decomposing learned representations into a predefined set of semantic components. Given an embedding z and a set of k semantic factors of interest, ConDe identifies k vectors with the same dimensionality as z, such that each of them is trained to capture a single factor, and their combined sum along with an additional residual can reconstruct z. Architecture and optimization details are provided in Appendices A–C. All experiments were run on an NVIDIA L40 with 48GB of VRAM.

Figure 1 illustrates ConDe's decomposition module. ConDe first estimates a residual component r intended to retain information not assigned to any factor. An estimated semantic subrepresentation z_{sem} is then processed by k parallel heads, each producing a component s_i associated with one fac-

tor. These components are trained to capture their designated semantics while jointly reconstructing the input through the fixed additive composition. This construction exposes selected semantic information in directly recombinable components while preserving unspecified variation in the residual.

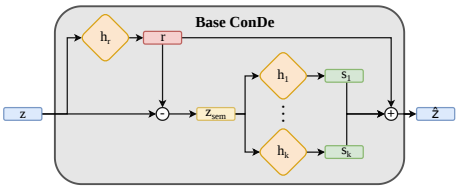

Figure 1: Architecture of Base ConDe.

Formally, let $z \in \mathbb{R}^d$ denote an input representation and let $v_1, \ldots, v_k$ denote the semantic factors of interest. ConDe produces corresponding vector components $s_1, \ldots, s_k \in \mathbb{R}^d$ and a residual $r \in \mathbb{R}^d$ such that $\hat{z} = r + \sum_{i=1}^{k} s_i$.

ConDe consists of k semantic heads $h_1, \ldots, h_k$ and one residual head h_r.We define a semantic subrepresentation $z_{\text{sem}} = z - r$, which serves as the input to the semantic heads: $s_i = h_i(z_{\text{sem}})$ for $i \in \{1, \ldots, k\}$. By operating directly on z_{sem}, we avoid issues that could arise from partial decomposition, allowing the semantic components to be more easily controlled.

The residual head is implemented as an element-wise gate: $r = \sigma(f_\phi(\text{LayerNorm}(z))) \odot z$, where f_ϕ is an MLP, σ is the sigmoid function, and $\odot$ denotes element-wise multiplication. This parametrization constrains r to be a gated subcomponent of z, preventing the residual head from introducing information absent from the input representation. The semantic heads remain unconstrained MLPs, allowing the components to overlap temporarily during early optimization before specialization emerges.

By definition, ConDe cannot alter the geometry of z, and can only separate components that are already disentangled in the original representation. To address this without leaving our post-hoc setting, we employ an INN (Dinh et al., 2016) to learn a bijective transformation $z' = f_\theta(z)$, with $z = f_\theta^{-1}(z')$ recovering z exactly. ConDe then operates on z', with f_θ learning it by propagating the gradients of $\mathcal{L}_{ConDe}$; the coverage loss $\mathcal{L}_{cov}$ is instead computed on the original representation z to stabilize the learned components. Any other learnable transformation could serve the same purpose, including one trained jointly with the encoder end-to-end.

3.1 Learning Objectives

To obtain a meaningful semantic decomposition, ConDe combines four objective functions, illustrated in Figure 2. First, a contrastive loss serves as a form of relational supervision to ensure each component captures the intended factor. The decomposition and coverage losses then promote completeness by encouraging the semantic components to collectively reconstruct z_{sem} and, ultimately, as much of z as possible. Finally, the residual loss minimizes statistical dependence between the residual and semantic components, thus pushing semantic information away from the residual.

As a contrastive term, we employ a triplet loss (Schroff et al., 2015) for its flexibility with varying numbers of positive samples in a batch:

$$\mathcal{L}_{triplet} = \sum_{i=1}^{k} \max(0, d_{\cos}(s_i^a, s_i^p) - d_{\cos}(s_i^a, s_i^n) + \alpha),$$

where $d_{\cos}$ is the cosine distance, $\alpha = 0.5$ is the margin the representations must respect, and the superscripts a, p, and n are used to denote anchor, positive and negative samples. This objective can easily be minimized by encoding only enough information to satisfy the margin, leaving other aspects of v_i in another component. We therefore introduce two cross-reconstruction terms to ensure the completeness of the representation.

The decomposition and coverage losses encourage the semantic components obtained from positive examples to reconstruct the anchor semantic subrepresentation and full representation, respectively:

$$\mathcal{L}_{decomp} = \text{MSE}(\sum_{i=1}^{k} s_i^p, \text{sg}(z_{\text{sem}}^a)), \qquad \mathcal{L}_{cov} = \text{MSE}(\sum_{i=1}^{k} s_i^p, \text{sg}(z^a)).$$

Here $\text{sg}(\cdot)$ denotes the stop-gradient operator. Reconstructing z_{sem}^a from these positive components encourages each head to preserve the information consistently associated with its designated factor

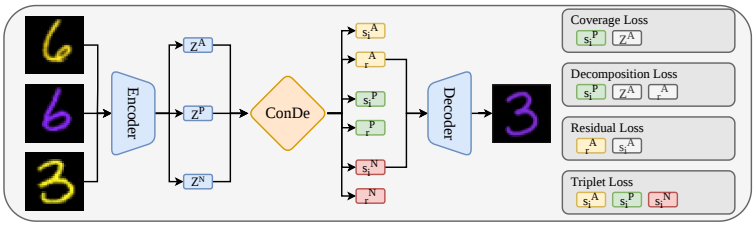

Figure 2: ConDe training procedure. ConDe receives a triplet of samples containing positive and negative pairs for each factor. The obtained components may be shuffled across different examples to alter the corresponding factors in the decoder's output.

while discarding information that varies across the pair. Whereas $\mathcal{L}_{decomp}$ enforces consistency with the current semantic subrepresentation, $\mathcal{L}_{cov}$ encourages the semantic components to recover as much factor-related information from the full representation as possible. Together, these terms discourage the residual from becoming a shortcut that absorbs information attributed to the specified factors. The stop-gradient operator prevents the target representations from discarding information merely to match the current decomposition.

Finally, we encourage the residual to remain independent of the semantic components by minimizing the Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2005):

$$\mathcal{L}_{res} = \sum_{i=1}^{k} \mathrm{HSIC}(r^a, \mathrm{sg}(s_i^a)).$$

Applying stop-gradient to s_i^a ensures that a semantic component is not penalized for containing factor information that has not yet been removed from r. We choose HSIC for its capability to penalize nonlinear relationships which could otherwise remain in r.

The complete ConDe objective sums the triplet, decomposition, coverage, and residual losses, each weighted by 1 by default unless stated otherwise.

3.2 Soft Pairing

The presented formulation assumes that positive pairs define the intended semantic factor directly. We refer to this setting as *strict pairing*. Let x^a and x^{p_i} be samples that share factor v_i, while their remaining factors vary independently, and let x^{n_i} differ from x^a in v_i. Under strict pairing, the information consistently shared between x^a and x^{p_i}, but not between x^a and x^{n_i}, directly defines v_i. For example, different renderings of the same digit in MNIST may share the same label but differ in handwriting style. As a result, treating these samples as positive pairs for the digit factor would encourage the corresponding component to represent class itself rather than sample-specific style.

Soft pairing relaxes the assumption that positive samples share identical factor information, so that $v_i^a \neq v_i^{p_i}$ in general; the positive component can therefore no longer replace the anchor component in the decomposition and coverage objectives, which are instead computed from the anchor components alone. The relational constraint is retained through a smooth triplet loss:

$$\mathcal{L}_{\text{soft}} = \sum_{i=1}^{k} \mathrm{softplus}\left(\frac{d_{\cos}(s_i^a, s_i^p) - d_{\cos}(s_i^a, s_i^n) + \alpha}{\tau}\right).$$

This objective preserves the positive–negative ordering of the standard triplet loss while replacing its hard margin with a smooth transition, allowing the network to align positive and anchor samples as closely as is reasonable for a given factor. The temperature τ controls this transition. Strict and soft pairing may be chosen independently for each factor.

4 Experiments

We evaluate both the quality and practical utility of semantic decomposition. We first assess whether ConDe isolates the intended factors under controlled settings, then examine what these components

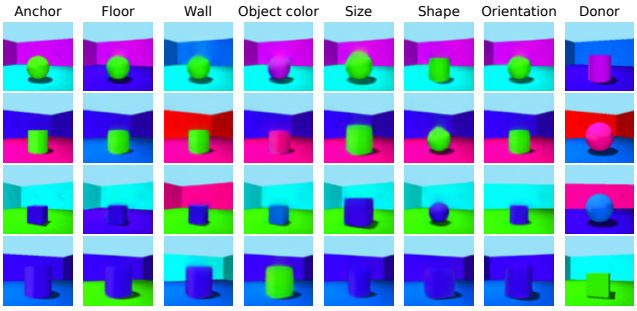

Figure 3: Reconstructions after swapping each factor in 3dshapes using ConDe.

enable in downstream applications, including style transfer and domain generalization. Unless otherwise stated, all experiments are run using invertible networks applied to frozen pretrained representations of regular autoencoders. Metric definitions and seismic experiment details are provided in Appendices E and F.

4.1 Evaluation metrics

Classical disentanglement metrics were designed primarily for dimension-wise representations. The Mutual Information Gap (MIG) measures whether each generative factor is concentrated in a single latent coordinate, while DCI summarizes whether latent coordinates are disentangled, complete, and informative with respect to known factors (Chen et al., 2018; Eastwood & Williams, 2018). These assumptions do not directly match ConDe, which follows a vector-wise approach to partial decomposition. We therefore use MIG and DCI only through adaptations, reported as secondary comparison metrics rather than as the main evaluation criterion.

We evaluate three properties of semantic latent decomposition. *Informativeness* measures the predictive information about v_i available in its assigned component s_i, using entropy-normalized cross-entropy for classification and clipped R^2 for regression. *Exclusivity* measures leakage by probing progressively larger PCA prefixes of the aggregated non-target components; high scores indicate that v_i is concentrated in s_i. *Invariance* measures the stability of s_i across positive pairs, relative to the distance between their complete representations. We additionally report Block DCI, which constructs the DCI importance matrix over components rather than coordinates, and Group MIG, which measures whether each factor's mutual information is concentrated in one component. At the decoder level, negative swap measures whether an image-level pretrained classifier can correctly predict the target factor when s_i is replaced with the corresponding component from a negative sample, while positive reconstruction tests whether replacing components with positive-pair counterparts preserves the original output.

4.2 Datasets

We evaluate ConDe across multiple datasets spanning vision, audio, and geological imaging. For controlled decomposition, we use 3D Shapes as the primary disentanglement benchmark. Its complete factorial structure provides exact labels and strict pairs for all generative factors, but also makes the task comparatively simple, causing several metrics to saturate. We therefore introduce two additional benchmarks that target limitations not captured by this setting.

Colored MNIST evaluates soft pairing. Each image is assigned one of eight color prototypes, later perturbed with a small random variation. Samples sharing a color label therefore exhibit similar, but not identical, colors, and, similarly, samples sharing the same digit label vary in handwriting-style, thus requiring the learned component to preserve variation beyond the discrete label.

SynBeeps is a synthetic audio dataset designed to evaluate vector-wise partial decomposition. It contains $60{,}000$ one-second signals, each composed of one to three beeps and generated from six semantic factors: pitch, timbre, loudness, reverberation, excitation, and duration. Additional unmodeled variations, including equalization and background noise, must remain in the residual rather than be assigned to the specified components.

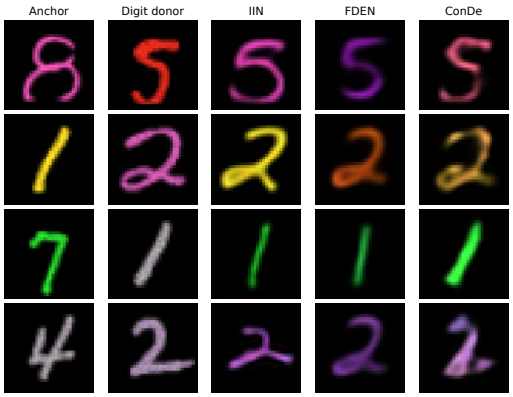

Figure 4: Digit-factor swaps for the three models.

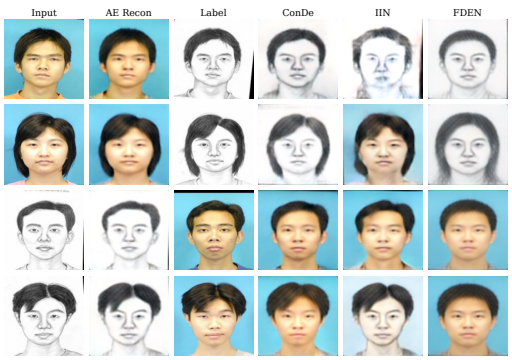

Figure 5: CUFS style transfer. Top rows: photo to sketch; bottom rows: sketch to photo.

	Representation					Output	
Method	Inf.	Exc.	Inv.	DCI	GMIG	N-S	P-R
3D Shapes							
ConDe	**1.00**	**0.99**	**0.96**	0.99	0.79	**0.97**	**0.99**
FDEN	**1.00**	**0.99**	0.87	**1.00**	**0.91**	0.85	**0.99**
IIN	0.96	0.74	0.22	0.47	0.69	0.65	**0.99**
SynBeeps							
ConDe	0.95	**0.99**	**0.97**	**0.97**	**0.89**	0.21	**0.97**
FDEN	**0.97**	0.98	0.77	**0.97**	0.79	**0.78**	0.96
IIN	0.68	0.91	0.08	0.21	0.14	0.00	0.85
Colored MNIST							
ConDe	0.95	0.87	0.46	0.70	0.56	**0.74**	**0.74**
FDEN	**0.99**	**0.99**	**0.67**	**0.99**	**0.98**	0.49	0.67
IIN	0.54	**0.99**	0.05	0.23	0.00	0.00	0.68

Table 1: Mean results across evaluation folds on three decomposition benchmarks. Inf., Exc., and Inv. denote informativeness, exclusivity, and invariance; N-S and P-R denote negative swap and positive reconstruction. All metrics lie in $[0, 1]$, and higher is better. All 95% confidence intervals had half-widths below 0.01 and are omitted for readability.

4.3 Disentanglement Analysis

We compare ConDe with FDEN and IIN using identical pretrained representations. Table 1 reports representation and decoder-level metrics, while Figures 3 and 4 illustrate representative semantic manipulations.

Both ConDe and FDEN substantially outperform IIN on the strict-pairing benchmarks. While 3D Shapes largely saturates, SynBeeps provides a more challenging comparison, though both models achieve strong results. FDEN achieves a better negative-swap score, whereas ConDe has higher invariance and Group MIG. FDEN trains its components to predict factor labels directly, whereas ConDe uses factor-matched pairs. Negative swap decodes a component-swapped representation and scores whether an output classifier recovers the donor label, so FDEN's objective is more directly aligned with this metric. This alignment may contribute to its SynBeeps advantage, although the score alone does not establish the cause. Colored MNIST instead evaluates soft pairing. Although FDEN again achieves strong quantitative scores, these metrics primarily reflect label recovery, which do not perfectly align with the soft paired factors. Figure 4 shows that FDEN often reconstructs an average digit prototype rather than taking the donor's handwritten appearance. ConDe preserves both digit identity and sample-specific handwriting, demonstrating that soft pairing captures semantic information beyond that tied to labels.

Takeaways. The experiments show that the methods target different notions of disentanglement. FDEN is highly effective when semantic factors are completely specified by labels, reflecting its direct supervision. ConDe retains this performance under strict pairing while naturally extending to soft relational supervision, where preserving instance-specific information becomes essential. IIN, in contrast, exhibited unstable optimization and consistently weaker results.

4.4 Ablation Study

On SynBeeps, we ablated each loss term and varied INN depth. Removing any loss reduces at least one metric; removing the contrastive term lowers five of seven scores, including a 0.10 drop in Group MIG (Table 6). The no-INN variant performs poorly, while increasing depth improves negative-swap score from 0.00 with two layers to 0.30 with 48; most representation metrics remain stable across depths. Further results appear in Appendix D.

4.5 Cross-Domain Style Transfer

While the previous experiments evaluate representation quality directly, we now investigate whether the learned decomposition improves downstream tasks that explicitly rely on controllable semantic representations. We start with direct semantic editing for style transfer. Using the CUHK face sketch dataset (CUFS) (Zhang et al., 2011), we decompose each latent representation into identity and style (photo versus sketch), then perform translation by replacing the style component while preserving identity. All methods operate on the same frozen pretrained autoencoder, requiring no task-specific generators or refinement networks. Figure 5 presents representative translation results. ConDe consistently preserves facial identity while translating between photos and sketches. IIN produces a successful style transfer in only one of the four examples, with the remaining outputs retaining the source modality or exhibiting reconstruction artifacts or identity distortions. FDEN produces visually sharp sketches, but its outputs often converge toward average faces, suppressing identity-specific characteristics such as facial proportions and hairstyle. This behavior mirrors the Colored MNIST results, where direct supervision encouraged recovery of semantic labels while sacrificing instance-specific information.

Takeaways. Although CUFS is a small dataset and the evaluated methods are not designed as dedicated image translation models, the results demonstrate that semantic decomposition can enable meaningful style transfer through direct latent recombination. In particular, ConDe preserves identity while modifying the desired semantic factor, showing that factor-specific editing can be achieved directly on frozen representations without introducing task-specific generative architectures.

4.6 Case Study: End-to-End Timbre Transfer

We next evaluate an end-to-end variant of ConDe on music timbre transfer, a more challenging real-world setting where we must modify instrument timbre while preserving the underlying musical content. ConDe is trained jointly with an audio autoencoder to find structure and timbre components. Transfer is performed by replacing the timbre component with that of a reference audio. We compare against the autoencoder-based SS-VQ-VAE (Cífka et al., 2021) and the diffusion-based AFTER (Demerlé et al., 2024). All methods are trained on a paired version of the ASAP dataset (Foscarin et al., 2020), in which each piece was rendered for four different instruments: piano, xylophone, vibraphone and harp.

Method	Acc. ↑	Time ↓	DTW ↓	Onset ↑
SS-VQ-VAE	0.246	0.020s	0.292	0.063
AFTER	0.965	0.701s	0.290	0.643
ConDe	**0.975**	**0.010s**	**0.077**	**0.787**
Input audio	0.974	–	–	–

Table 2: Timbre-transfer results. Accuracy measures recovery of the reference instrument post-transfer; DTW and onset correlation measure similarity between the original source and transferred audio.

Takeaways. Table 2 shows that ConDe matches the instrument-classification performance of the original audio while preserving musical structure more accurately than both baselines. Unlike diffusion-based approaches, transfer is achieved through a single latent recombination and decoding

step, resulting in substantially lower inference time. These results demonstrate that semantic decomposition is not only useful for controlled benchmark manipulations, but also enables high-quality semantic editing in a realistic application through direct manipulation of learned representations.

4.7 Domain Invariance

We next evaluate whether semantic decomposition can induce domain invariance within a standard classifier. We follow the DomainBed evaluation protocol (Gulrajani & Lopez-Paz, 2020) evaluating on the held-out domain. ConDe is inserted into the classifier bottleneck and trained end-to-end with a single soft-paired component representing the environment; classification is performed from the residual.

Method	PACS	VLCS	OfficeHome	TerraInc.	Avg.
ERM	84.2(0.1)	77.3(0.1)	67.6(0.2)	47.8(0.6)	69.2
DANN	83.6(0.4)	78.6(0.4)	65.9(0.6)	46.7(0.5)	68.7
CORAL	86.2(0.3)	78.8(0.6)	68.7(0.3)	47.6(1.0)	70.3
SWAD	88.1(0.1)	**79.1**(0.1)	70.6(0.2)	50.0(0.3)	72.0
ERM++	**89.8**(0.3)	78.0(0.1)	**74.7**(0.0)	**51.2**(0.3)	**73.4**
ConDe	84.8(0.4)	76.7(0.3)	71.5(0.1)	46.5(0.5)	69.9

Table 3: Target-domain accuracy using training-domain validation for model selection. Values are mean (95% CI).

Takeaway. ConDe slightly outperforms ERM on average and obtains a comparatively strong result on OfficeHome, showing that isolating environmental information can yield a useful domain-invariant residual. However, it remains behind specialized domain-generalization methods. This experiment therefore establishes domain invariance as a possible use of semantic decomposition, rather than a primary application or state-of-the-art result for ConDe.

4.8 Case Study: Domain Generalization in Geology

We finally evaluate vector-wise disentanglement on a real-world domain generalization task using the DeepFault seismic dataset (Guo et al., 2025) for pixel-wise segmentation of geological faults. The original images form the source domain, while a target domain is constructed through structure-preserving appearance transformations. ConDe is trained end-to-end with a single semantic component corresponding to geological structure, which serves as the input for the segmentation decoder. We compare against an identical source-only model trained on the original images and a target-domain oracle trained directly on target data.

Method	F1 ↑	Prec. ↑	Rec. ↑
Source only	0.377	0.406	0.351
ConDe	0.517	0.430	0.650
Target oracle	0.585	0.522	0.665

Table 4: Fault segmentation performance on the target seismic domain.

Takeaways. Table 4 shows that ConDe consistently improves target-domain fault segmentation over the source-only baseline, indicating that the learned structure component retains information relevant to fault detection while suppressing domain-specific appearance variations. Appendix Figure 6 further illustrates this behavior: whereas test loss increasingly diverges on the source-only model, ConDe's structure component yields a more stable optimization throughout training. Although the oracle remains an upper bound obtained with target supervision, these results demonstrate that semantic decomposition provides a practical mechanism for improving robustness to real-world domain shifts without requiring target-domain labels. Further experimental details are in Appendix F.

5 Conclusion

We introduced ConDe, a weakly supervised framework that decomposes representations into additive semantic vectors and a residual, either post hoc through an invertible transformation or end to end. Across controlled benchmarks, ConDe achieves consistent decomposition under strict and soft pairing, while our evaluation reveals limitations of classical dimension-wise metrics. Image and music transfer and seismic domain generalization show that its components can be manipulated directly or isolated for downstream tasks.

AI USE STATEMENT

Generative AI tools were used to help refine the manuscript, find relevant bibliography, and implement code. The authors reviewed all AI-assisted material and take responsibility for the final paper and accompanying code.

REPRODUCIBILITY STATEMENT

Implementation and evaluation details are provided in the appendix. Anonymized code is available at `https://anonymous.4open.science/r/conde-sld-314D/`.

REFERENCES

Tal Barami, Nimrod Berman, Ilan Naiman, Amos H Hason, Rotem Ezra, and Omri Azencot. Disentanglement beyond static vs. dynamic: A benchmark and evaluation framework for multi-factor sequential representations. *Advances in Neural Information Processing Systems*, 38, 2026.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Michael Bereket and Theofanis Karaletsos. Modelling cellular perturbations with the sparse additive mechanism shift variational autoencoder. *Advances in Neural Information Processing Systems*, 36:1–12, 2023.

Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *Advances in neural information processing systems*, 29, 2016.

Pierre Charbonnier, Laure Blanc-Féraud, Gilles Aubert, and Michel Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on image processing*, 6(2): 298–311, 1997.

Junxiang Chen and Kayhan Batmanghelich. Weakly supervised disentanglement by pairwise similarities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3495–3502, 2020.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

Ondřej Cífka, Alexey Ozerov, Umut Şimşekli, and Gael Richard. Self-supervised vq-vae for one-shot music style transfer. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 96–100. IEEE, 2021.

Nils Demerlé, Philippe Esling, Guillaume Doras, and David Genova. Combining audio control and style transfer using latent diffusion. *arXiv preprint arXiv:2408.00196*, 2024.

Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5154–5163, 2020.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International conference on learning representations*, 2018.

Patrick Esser, Robin Rombach, and Bjorn Ommer. A disentangling invertible interpretation network for explaining latent representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9223–9232, 2020.

Francesco Foscarin, Andrew McLeod, Philippe Rigaux, Florent Jacquemard, and Masahiko Sakai. ASAP: a dataset of aligned scores and performances for piano transcription. In *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 534–541, 2020.

Marco Fumero, Luca Cosmo, Simone Melzi, and Emanuele Rodolà. Learning disentangled representations via product manifold projection. In *International conference on machine learning*, pp. 3530–3540. PMLR, 2021.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pp. 63–77. Springer, 2005.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Zhixiang Guo, Xinming Wu, Luming Liang, Hanlin Sheng, Nuo Chen, and Zhengfa Bi. Cross-domain foundation model adaptation: Pioneering computer vision models for geophysical data analysis. *Journal of Geophysical Research: Machine Learning and Computation*, 2(1): e2025JH000601, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016b.

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 (ICLR)*, 2017. URL `https://openreview.net/forum?id=Sy2fzU9gl`.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 424–434, 2019.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International conference on artificial intelligence and statistics*, pp. 2207–2217. PMLR, 2020.

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

Jack Klys, Jake Snell, and Richard Zemel. Learning latent subspaces in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

Dmytro Kotovenko, Artsiom Sanakoyeu, Sabine Lang, and Bjorn Ommer. Content and style disentanglement for artistic style transfer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4422–4431, 2019.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019a.

Francesco Locatello, Michael Tschannen, Stefan Bauer, Gunnar Rätsch, Bernhard Schölkopf, and Olivier Bachem. Disentangling factors of variation using few labels. *arXiv preprint arXiv:1905.01258*, 2019b.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International conference on machine learning*, pp. 6348–6359. PMLR, 2020.

Yotam Norman and Ron Meir. Unsupervised representation learning-an invariant risk minimization perspective. In *The Fourteenth International Conference on Learning Representations*, 2026.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Lei Sha and Thomas Lukasiewicz. Text attribute control via closed-loop disentanglement. *Transactions of the Association for Computational Linguistics*, 12:190–209, 2024.

Xin Wang, Hong Chen, Si'ao Tang, Zihao Wu, and Wenwu Zhu. Disentangled representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9677–9696, 2024.

Jee Seok Yoon, Myung-Cheol Roh, and Heung-Il Suk. A plug-in method for representation factorization in connectionist models. *IEEE Transactions on Neural Networks and Learning Systems*, 33(8):3792–3803, 2021.

Wei Zhang, Xiaogang Wang, and Xiaoou Tang. Coupled information-theoretic encoding for face photo-sketch recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 513–520. IEEE, 2011.

APPENDIX

A AUTOENCODER ARCHITECTURE

ConDe is not restricted to autoencoders; however, an encoder–decoder model is useful for evaluating whether latent interventions produce the intended changes in the reconstructed output. For these experiments, we use a convolutional autoencoder composed of residual blocks (He et al., 2016a).

The blocks follow a pre-activation ordering, with normalization and activation applied before the convolutional layers. This preserves a clean identity shortcut and improves optimization in deeper residual networks (He et al., 2016b). Motivated by ConvNeXt-style blocks (Liu et al., 2022), each residual block uses a single normalization layer at its input. We use Group Normalization and SiLU activations rather than Batch Normalization and ReLU. Group Normalization operates independently on each sample and is therefore better suited to embeddings produced by direct latent interventions, whose distributions may differ from those observed during standard reconstruction.

The encoder begins with a stem consisting of a convolution followed by Group Normalization and SiLU. The resulting features are processed by a sequence of residual blocks and projected through a linear layer into a d-dimensional embedding. A second linear layer maps the embedding to the decoder input space, after which a symmetric sequence of residual blocks reconstructs the input. Downsampling is performed with strided convolutions, whereas upsampling uses nearest-neighbor interpolation followed by convolution. The final decoder stage contains one additional upsampling operation to recover the input resolution and a final output convolution. All convolutional layers use

kernels of size 3. Network depths and channel widths vary by experiment and are reported with the corresponding experimental settings.

All autoencoders are trained with a Charbonnier loss (Charbonnier et al., 1997):

$$\sqrt{(x - \hat{x})^2 + \epsilon^2},$$

averaged over the output dimensions, with $\epsilon = 10^{-3}$. The loss behaves approximately linearly for large errors, reducing sensitivity to outliers, while remaining smooth and approximately quadratic near zero.

B ConDe and INN Architecture

B.1 Decomposition Heads

The residual mapping f_ϕ in the residual-head equation is a four-layer MLP. Layer Normalization is applied to z before the MLP, and the resulting gate is bounded by a sigmoid activation. Each semantic head h_i is an unconstrained two-layer MLP that receives z_{sem} and outputs a vector with the same dimensionality as z. The hidden dimensionality of all heads equals the input dimensionality, and SiLU activations are used throughout. No normalization is applied within the semantic heads.

Keeping all components in $\mathbb{R}^d$ allows them to be recombined through direct addition. The dimensionality of a component should therefore not be interpreted as the number of independent scalar factors assigned to it: each s_i is a vector-valued representation of one specified semantic factor.

B.2 HSIC Implementation

We implement the residual-independence objective with a multi-kernel variant of HSIC. Before computing the kernels, r and s_i are standardized independently along each dimension. We estimate a median-distance bandwidth for each representation from at most 512 samples and use their geometric mean as the shared base bandwidth σ_0.

For a batch of n samples, we use the biased empirical estimator

$$\text{HSIC}(r, s_i) = \frac{1}{(n-1)^2} \operatorname{tr}(KHLH),$$

where $K, L \in \mathbb{R}^{n \times n}$ are the kernel Gram matrices of r and s_i, respectively, $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is the centering matrix, and $\operatorname{tr}(\cdot)$ denotes the trace operator.

We construct each Gram matrix by averaging six RBF kernels with bandwidths $m\sigma_0$, where

$$m \in \left\{ \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 4 \right\}.$$

This multi-scale construction captures dependencies at different geometric scales.

B.3 Invertible Transformation

The post-hoc transformation f_θ contains four to twelve affine coupling layers, depending on the experiment. For an input partition (z_a, z_b), each layer applies

$$u_a = z_a, \qquad u_b = z_b \odot \exp(s(z_a)) + t(z_a),$$

where $s(\cdot)$ and $t(\cdot)$ are produced by a shared two-layer MLP. The hidden dimensionality is $d/2$, and SiLU activations are used between linear layers. The dimensions are reassigned to active and passive partitions through fixed random permutations between coupling layers, ensuring that every coordinate is transformed and cross-conditioned across the network.

We initialize the INN close to the identity transformation. The output of each coupling MLP is multiplied by a learned scalar gate initialized to zero. The final linear layer has zero bias and weights sampled from $\mathcal{N}(0, 10^{-4})$. We additionally constrain the scale output with a hyperbolic tangent before the exponential operation, reducing the risk of unstable scaling and exploding gradients during early training.

C Triplet Construction and Optimization

C.1 Triplet Construction

Each training item contains three samples (A, B, C), which are used jointly for all k semantic factors. Sample A is always the anchor. Sample B is selected as the positive for $\lfloor k/2 \rfloor$ factors and as the negative for the remaining factors; C assumes the complementary role. Thus, for every factor, one of B or C is positive and the other is negative.

This construction encourages the factors to vary independently across triplets and reduces reliance on correlations that are not part of the pairing definition. It is not required by the framework: positive and negative examples may instead be identified independently for each factor within a sufficiently large unstructured batch.

C.2 Autoencoder Optimization

Unless stated otherwise, the autoencoder is trained for 250 epochs with AdamW and a constant learning rate of 10^{-4}. Weight decay of 10^{-4} is applied only to the encoder. The decoder is left unregularized to prioritize reconstruction fidelity.

C.3 ConDe Optimization

ConDe is trained with AdamW and a OneCycle learning-rate schedule. The learning rate reaches a maximum of 10^{-4} after 6% of the optimization steps. Weight decay is set to 10^{-6} for all ConDe parameters, and gradients are clipped to a maximum norm of 5.0.

For end-to-end experiments, the encoder uses a maximum learning rate of 2×10^{-5} and weight decay of 10^{-5}. The decoder uses a learning rate of 10^{-5} without weight decay. Unless otherwise specified, all ConDe loss coefficients are set to 1. Experiment-specific architectures, attenuation strengths, margins, and deviations from these defaults are reported in the corresponding experimental sections.

D Ablation Study

We study individual ConDe design choices on SynBeeps. All ablations use the post-hoc setting and the same 512-dimensional autoencoder representation as the corresponding main experiment.

D.1 Invertible Transformation Depth

We first test how sensitive ConDe is to the depth of the invertible network. We compare a variant without the INN (zero coupling layers) with depths in $\{2, 4, 8, 12, 24, 48\}$ while holding all other hyperparameters fixed. Table 5 reports the obtained representation and output metrics.

	Representation					Output	
Layers	Inf.	Exc.	Inv.	DCI	GMIG	N-S	P-R
0	0.96	0.99	0.95	0.97	0.78	0.00	0.95
2	0.96	0.99	0.97	0.97	0.82	0.00	0.97
4	0.96	0.99	0.97	0.97	0.86	0.04	0.97
8	0.95	0.99	0.97	0.97	0.88	0.17	0.96
12	0.95	0.99	0.97	0.97	**0.89**	0.21	0.97
24	0.95	0.99	0.97	0.97	**0.89**	0.24	0.96
48	0.95	0.99	0.98	0.97	0.87	**0.30**	0.96

Table 5: Effect of invertible-transformation depth on SynBeeps; zero layers omit the INN. Inf., Exc., Inv., DCI, and GMIG denote representation-level metrics: informativeness, exclusivity, invariance, Block DCI, and Group MIG. N-S and P-R denote negative swap and positive reconstruction. Values are aggregate means under the protocols described in Appendix E. All 95% confidence-interval half-widths for the nonzero-depth runs are below 0.01 and are omitted; the zero-layer run is reported without an interval.

	Representation					Output	
Variant	Inf.	Exc.	Inv.	DCI	GMIG	N-S	P-R
Full ConDe	0.96	0.99	0.97	0.97	0.86	0.04	0.97
w/o Cont.	0.94	0.98	0.96	0.97	0.76	0.27	0.95
w/o Dec.	0.96	0.99	0.96	0.97	**0.91**	**0.36**	0.91
w/o Cov.	0.95	0.98	**0.99**	**0.98**	0.88	0.00	**1.00**
w/o Res.	0.96	0.99	0.98	0.97	0.88	0.00	0.98

Table 6: Effect of removing individual loss terms on SynBeeps with a four-layer invertible transformation. Column groups and abbreviations follow Table 5. Values are aggregate means under the protocols in Appendix E. All 95% confidence-interval half-widths are below 0.01 and are omitted.

Without the INN, informativeness remains 0.96 and exclusivity 0.99, but negative swap is 0.00 despite positive reconstruction of 0.95. The semantic heads can extract factor-predictive features, but without a learnable transformation they cannot reorganize the frozen embedding for intervention. The failed swaps are consistent with factor information remaining in the residual and overriding an exchanged component; these aggregate metrics do not directly measure residual factor content. Most representation metrics remain stable as depth increases. Group MIG rises from 0.78 without the INN to 0.89 with twelve or twenty-four layers, then decreases to 0.87 with forty-eight. Negative swap rises from 0.00 to 0.30 across the depth sweep, whereas positive reconstruction stays near 0.95–0.97. These results suggest that representation probes and decoded swaps capture different properties. A shallow transformation can make the designated factor recoverable from its component, yet a successful decoded intervention also requires the additive recombination to remain compatible with the pretrained decoder. We retain twelve layers for 3D Shapes and SynBeeps and four for Colored MNIST, which requires less computation due to its simplicity. Although greater depths would lead to better results, they come at the cost of added computational resources, which were not necessary for the benchmarks tested.

D.2 Loss Component Ablation

ConDe combines a contrastive triplet loss $\mathcal{L}_{triplet}$, a decomposition loss $\mathcal{L}_{decomp}$, a coverage loss $\mathcal{L}_{cov}$, and a residual-independence loss $\mathcal{L}_{res}$. We remove one term at a time ($\lambda_i = 0$) and retrain on SynBeeps with all other hyperparameters fixed. Every variant, including the full-model reference, uses four coupling layers. This controls for the depth effects examined in Appendix D.1. Table 6 reports the results.

Removing the contrastive loss causes the broadest decline. Informativeness and Group MIG undergo their largest decreases in Table 6. Without the relational signal, the objective no longer specifies which factor each head should represent. The remaining completeness terms can still encourage joint reconstruction, but they do not identify the assignment between heads and factors.

Removing either coverage or residual independence leaves the representation metrics close to those of the full model. Some even increase slightly. Negative swap, however, falls to 0 in both cases. This contrast is consistent with residual leakage: probes can find clean factor information in the semantic components even when the residual retains redundant information that limits the effect of a swap. Coverage discourages the residual from absorbing information needed to reconstruct the specified factors, while residual independence penalizes dependence between r and each s_i. The output metrics expose this shortcut because they preserve the anchor residual during an intervention.

Removing the decomposition loss raises negative swap from 0.04 to 0.36, the highest value in Table 6. It also lowers positive reconstruction from 0.97 to 0.91. Because coverage already imposes completeness relative to the full representation z, removing the z_{sem}-relative term may permit larger component changes at the expense of reconstruction fidelity. We retain the decomposition loss because controlled generation requires both effective negative swaps and stable positive swaps. The result nevertheless indicates that the two completeness terms have distinct effects.

E METRICS

We evaluate the decomposition at two levels. Representation metrics operate directly on the semantic components, whereas output metrics measure the effect of controlled component swaps after decoding. Unless noted otherwise, reported scores are unweighted averages over factors.

E.1 REPRESENTATION METRICS

E.1.1 PREDICTIVE INFORMATION AND PROBES

Several metrics use probes to measure how readily a factor v_i can be recovered from a representation. For classification, we normalize probe cross-entropy by the empirical entropy of the test-fold labels:

$$\mathrm{PI}(q, v_i) = \mathrm{clip}\left(1 - \frac{\mathrm{CE}(v_i, \hat{v}_i)}{H(v_i) + \epsilon}, 0, 1\right).$$

A score of zero indicates no improvement over the empirical label prior, whereas one indicates perfect probabilistic prediction. For regression, predictive information is the test R^2 clipped to $[0, 1]$. This definition places classification and regression on a common bounded scale. It also avoids comparing raw accuracies across tasks with different numbers of classes.

All probe inputs are standardized using statistics fitted on the training fold. Linear classification probes use multinomial logistic regression with $C = 1$ and at most 1000 iterations. Linear regression probes use ridge regression with $\alpha = 1$. We use five shuffled stratified folds for classification and five shuffled folds for regression, both with seed 42. Approximate 95% confidence intervals are the mean plus or minus 1.96 standard errors across folds.

E.1.2 INFORMATIVENESS

For each factor v_i, informativeness is $\mathrm{PI}(s_i, v_i)$ under the linear probe defined above. Absolute informativeness averages this value over factors.

E.1.3 EXCLUSIVITY

Exclusivity measures how much information about v_i remains in the semantic components other than s_i. We aggregate these non-target components using each method's composition: addition for ConDe and concatenation for partition-based baselines. We standardize the resulting complement and fit PCA using only the training fold. Linear probes are then trained on progressively larger PCA prefixes. The prefixes correspond to cumulative explained-variance thresholds spaced by 0.01; duplicate dimensionalities are removed, and the full prefix is always included.

Let x_j be the cumulative explained variance and p_j the predictive information at prefix j, with $(x_0, p_0) = (0, 0)$. Leakage is the right-step area

$$A_i = \sum_{j=1}^{m} (x_j - x_{j-1}) p_j,$$

and exclusivity is $E_i = 1 - A_i$. The right-step integral assigns each predictive-information value only after its corresponding group of principal components has been added. We average E_i over factors and five folds.

E.1.4 INVARIANCE

For each factor v_i, we sample 5000 pairs that agree on v_i and differ in at least one other labeled factor. We measure root-mean-square distance over the latent coordinates. For a sampled pair (a, p),

$$I_i(a, p) = \mathrm{clip}\left(1 - \frac{\mathrm{RMS}(s_i^a, s_i^p)}{\mathrm{RMS}(z^a, z^p) + \epsilon}, 0, 1\right).$$

We first average over pairs and then over factors. To estimate variation, we repeat pair sampling five times with seeds $42, \ldots, 46$.

E.1.5 BLOCK DCI

Following Barami et al. (2026), we adapt DCI from individual coordinates to semantic blocks. For each factor, we train an Extremely Randomized Trees probe on the concatenated semantic blocks and sum feature importances within each block. The resulting block-by-factor importance matrix defines modularity, compactness, and explicitness. These are block analogues of disentanglement, completeness, and informativeness. We use five folds, 512 trees, a maximum depth of 7, square-root feature subsampling, and at least two samples per leaf. The residual is excluded. The aggregate Block DCI score is the harmonic mean of the three properties.

E.1.6 GROUP MIG

Following Sha & Lukasiewicz (2024), we adapt MIG to vector-valued components. Each component is standardized and reduced with PCA to retain 95% of its variance. We estimate mutual information between the reduced continuous component and each discrete factor with a k-nearest-neighbor estimator using $k = 10$. The estimate is normalized by the empirical factor entropy. For factor v_i,

$$\mathrm{GMIG}_i = \frac{\mathrm{MI}(s_i; v_i)}{H(v_i)} - \max_{j \neq i} \frac{\mathrm{MI}(s_j; v_i)}{H(v_i)}.$$

To keep every table metric higher-is-better, we map this gap from $[-1, 1]$ to $[0, 1]$ using $(\mathrm{GMIG}_i + 1)/2$. We average over factors and 20 random subsamples of at most 20,000 observations. Because high-dimensional mutual-information estimates depend on sample size, PCA, scaling, and neighborhood size, we treat Group MIG as a secondary diagnostic.

E.2 OUTPUT METRICS

For each factor, we sample 5000 anchor–positive–negative triples. The positive shares the anchor's target-factor label but differs in at least one other factor. The negative has a different target-factor label. We replace only the target component and retain the anchor residual and all other anchor components.

Positive reconstruction. Let $x_a = D(z_a)$ denote the unmodified autoencoder reconstruction, $x^{(i)}_{a \leftarrow p}$ the reconstruction after replacing s^a_i with s^p_i, and $x_{\pi(a)} = D(z_{\pi(a)})$ the reconstruction of a random non-anchor observation. For each pair,

$$\mathrm{PR}_i = \mathrm{clip}\left(1 - \frac{\mathrm{MSE}(x_a, x^{(i)}_{a \leftarrow p})}{\mathrm{MSE}(x_a, x_{\pi(a)}) + \epsilon}, 0, 1\right).$$

The reported positive-reconstruction value is therefore a normalized, higher-is-better score rather than raw MSE.

Negative swap. After replacing s^a_i with s^n_i, a separately trained output-level classifier predicts the negative donor's label. We compute the entropy-normalized predictive information used for representation probes and divide it by the classifier's predictive information on the unmodified negative reconstruction:

$$\mathrm{NS}_i = \mathrm{clip}\left(\frac{\mathrm{PI}(x^{(i)}_{a \leftarrow n}, v^n_i)}{\mathrm{PI}(D(z_n), v^n_i) + \epsilon}, 0, 1\right).$$

This normalization reduces the influence of limitations in the output-level evaluator.

For output metrics, we estimate 95% intervals from 1000 bootstrap resamples with seed 42. Reported values are macro averages over factors.

F SEISMIC EXPERIMENT DETAILS

All models were trained from scratch using the same encoder–decoder architecture. The encoder contains four stages with channel widths $[64, 128, 256, 512]$ and $[3, 3, 19, 3]$ residual blocks, respectively, and produces a 1024-dimensional representation. Intermediate layers use SiLU activations

and group normalization, while the final encoder and decoder layers use identity activations. Training used Adam with a learning rate of 10^{-4}, no weight decay and a batch size of 1024. ConDe is trained for 251 epochs. The fault-segmentation model is subsequently trained for 50 epochs using ConDe's structure representations. Figure 6 shows this second training stage.

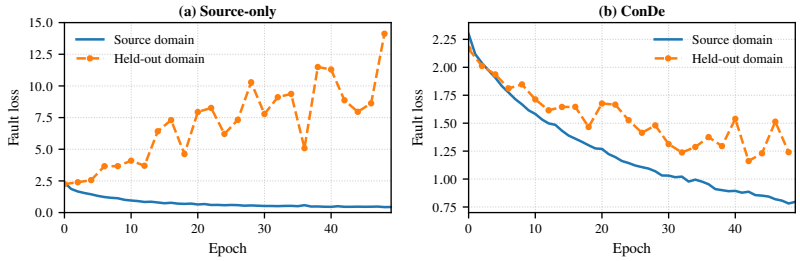

Figure 6: Train and test losses for a regular fault segmentation model, and one trained on ConDe's structure vector.

Domain construction and pairing. The target domain and the positive structure pairs used to train ConDe were constructed with the same augmentation pipeline. The transformations were chosen to alter seismic texture, resolution, and frequency content without changing the fault geometry. Every augmented image received an initial Gaussian blur with kernel size 15 and standard deviation sampled uniformly from $[1.2, 2.5]$. Fourier phase noise was applied with probability 0.30 and magnitude sampled from $[0.2, 0.4]$, while Fourier amplitude noise was applied with probability 0.75 and magnitude sampled from $[0.5, 0.8]$. Fourier amplitude smoothing was applied with probability 0.55, using a standard deviation sampled from $[0.8, 1.5]$. A frequency filter was applied with probability 0.70, with lower and upper thresholds sampled from $[40, 60]$ and $[80, 100]$, respectively. Phase smoothing, horizontal flipping, and scale-and-crop were disabled.

ConDe was trained end to end with one semantic component representing geological structure. Two independently augmented versions of the same seismic section formed a positive structure pair. The encoder, decoder, and ConDe decomposition were optimized jointly; no pretrained autoencoder was used. The reconstruction objective combined mean absolute error, a gradient-matching term weighted by 0.1, and an L2 penalty of 10^{-5} on the latent representation.

Fault segmentation. The segmentation decoder used only the ConDe structure component. It was trained using labeled source-domain samples and evaluated directly on the target domain. The source-only baseline used the same architecture but was trained end to end on the original source domain without decomposition. Fault segmentation was optimized with the sum of weighted binary cross-entropy and focal Tversky loss. The binary cross-entropy used a positive-class weight of 80 to compensate for the sparsity of fault pixels. The focal Tversky term was weighted by 0.4 and used $\alpha = 0.7$, $\beta = 0.3$, $\gamma = 1.33$, and a smoothing constant of 1.

Qualitative comparison. Figure 7 compares predictions from ConDe and the source-only model on fixed target-domain examples. The same samples are shown for both methods, including fault-free images and representative examples containing simple, branching, and low-contrast faults.

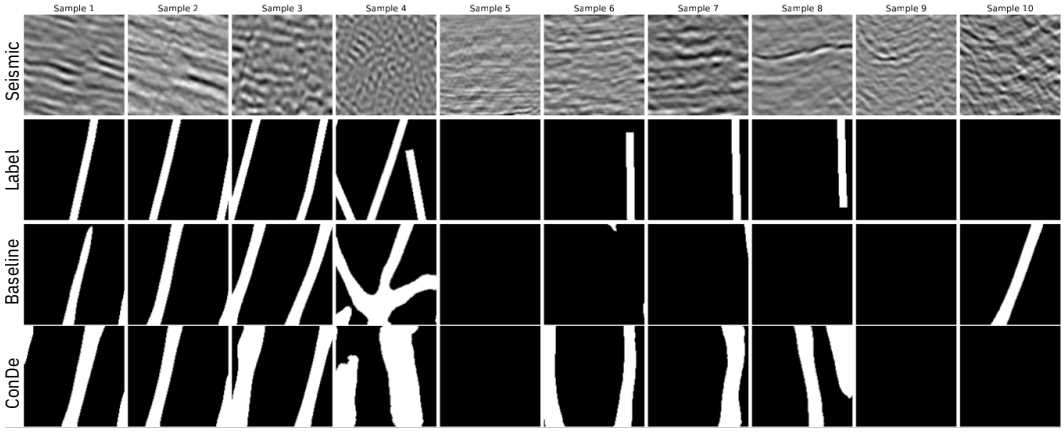

Figure 7: Fault-segmentation predictions on the target seismic domain. Rows show the seismic inputs, ground-truth masks, predictions from the end-to-end model trained only on the original source domain, and predictions obtained using the ConDe structure component. Columns correspond to fixed target-domain test samples.