LATTICE: LATENT IMAGINATION FOR COMPOSITIONAL EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep networks often fail on new combinations of familiar concepts, such as a blue triangle after seeing blue objects and triangles only separately, even in simple synthetic settings. Handling such combinations, an ability known as compositional generalization, is a hallmark of human intelligence and essential because the space of combinations grows exponentially while data does not. Most prior work pursues it through disentanglement, which attempts to isolate the attributes, yet evidence on whether disentanglement helps is mixed and on its own it has not solved the problem. We therefore propose to also train models explicitly to compose attributes. We introduce **LATenT Imagination for Compositional Embeddings (LATTICE)**, a training procedure that makes composition a task through analogy: just as king $-$ man $+$ woman $\approx$ queen in word embeddings, a blue triangle should be obtained from a blue circle by applying the change that turns a red circle into a red triangle. Given three training images that form a 2×2 attribute grid with a fourth, the model must imagine the representation of the fourth by this vector analogy. On six compositional generalization benchmarks, LATTICE raises the average accuracy of lightweight backbones by a relative 17–25% without adding parameters, letting one surpass a 3–6$\times$ larger backbone designed for this problem. It also makes representations measurably more parallel, even on unseen combinations: changing an attribute moves the representation in the same direction regardless of the others, a geometry also observed in the brain. Backbones that already provide this geometry gain little from LATTICE. Ablations show that each of its components is needed. We complement these results with a theoretical analysis.

1 INTRODUCTION

Deep learning owes much of its success on perceptual tasks to learning useful features directly from data (LeCun et al., 2015; Krizhevsky et al., 2012; He et al., 2016). Yet deep networks still struggle with *compositional generalization*, the ability to handle new combinations of familiar concepts and a hallmark of human intelligence (Fodor & Pylyshyn, 1988; Lake & Baroni, 2018): after seeing a red circle, a red triangle and a blue circle, a compositional learner should also recognize a blue triangle. This matters because the space of combinations grows exponentially while datasets do not, so a model will inevitably meet familiar values that never co-occurred in training. Yet widely used models fail at it even in deliberately simple synthetic settings (Camposampiero et al., 2025; Schott et al., 2022; Montero et al., 2021). These failures are telling because such settings are controlled: with every factor labeled, no confounders, and the unseen combinations known exactly, errors reflect a lack of compositionality itself. We discuss the limitations of this choice in Appendix A.

What would a model need in order to succeed? We argue that robust compositional generalization requires two complementary ingredients. First, a model must learn the basic building blocks, that is, the attributes or primitives that encode relevant properties of the world, such as shapes and colors. Second, it must learn the compositional rules that govern how these primitives can be recombined in meaningful ways to represent higher-level concepts. Once a model has learned meaningful attributes and valid composition rules, it can, in principle, imagine new configurations of them, including those not present during training, thereby enabling systematic reasoning.

Most prior work concentrates on the first ingredient, through disentanglement objectives (Higgins et al., 2017; Mathieu et al., 2016; Locatello et al., 2019) or architectures that separate attributes,

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.

such as *Attribute Invariant Networks* (AIN; Camposampiero et al., 2025). Evidence on whether disentanglement helps is mixed (Locatello et al., 2019; Montero et al., 2021; Schott et al., 2022), and on its own it has not solved the problem: identifying the parts is not the same as learning to recombine them, and standard supervision can favor memorizing the combinations seen in training (Zhang et al., 2017).

In this work we follow the principle behind Matching Networks (Vinyals et al., 2016), if compositionality or imagination are required at test time, they should be practiced during training. Specifically, we propose to train compositionality through *imagination*: the model imagines the representation of one combination from those of related ones, and is corrected by the representation of the real image. Our approach, **LATenT Imagination for Compositional Embeddings (LATTICE)**, implements imagination by analogy, much as word embeddings complete king $-$ man $+$ woman $\approx$ queen (Mikolov et al., 2013): a blue triangle should be obtained from a blue circle by applying the change that turns a red circle into a red triangle.

At every training step, LATTICE draws four training images whose attribute values form a 2×2 grid and asks the model to imagine the representation of the fourth from the other three (Fig. 1). Like Joint Embedding Predictive Architectures (JEPA; LeCun, 2022; Assran et al., 2023; Bardes et al., 2024), it predicts in an abstract representation space rather than in pixel space: the fourth image exists, so its own encoder representation is the target, and no pixel-level detail irrelevant to the attributes has to be reproduced. Unlike JEPA, which predicts masked parts of the same input, LATTICE predicts one sample from others, which is what compositionality requires.

Imagining by a fixed analogy has a geometric consequence. Since the rule cannot adapt, the model can only imagine correctly by reshaping its encoder so that each attribute change moves the representation in the same direction regardless of the other attributes: the four representations of a grid form a parallelogram (Fig. 1, right). Such *parallel* representations have been found in the brain, where they are associated with generalization to new conditions, and can be measured with the parallelism score (P-score; Bernardi et al., 2020). This geometry is also why imagination can reach combinations never seen in training: their representation is determined by the seen ones. Training cannot guarantee that an unseen image is encoded there, since the loss never observes one; that depends on whether the encoder isolates the attributes of new images, the first ingredient, which architectures such as AIN are designed to provide. Disentanglement thus supplies the parts, and LATTICE trains their recombination.

To test this complementarity, we evaluate LATTICE on six compositional generalization benchmarks with three backbones of increasing built-in disentanglement: ResNet-18, AIN, and Essentially Disentangled (ED; Camposampiero et al., 2025), a backbone designed for compositional generalization that builds disentanglement into its structure with a separate encoder for every attribute, at the cost of 3–6$\times$ more parameters. Consistent with this view, LATTICE makes representations more parallel, also on unseen combinations, and raises the accuracy of ResNet-18 and AIN by a relative 25% and 17%, letting AIN surpass ED at a fraction of its parameters; on ED, whose representations are already the most parallel by design, it adds little. We also compare and combine LATTICE with Compositional Risk Minimization (CRM; Mahajan et al., 2025), a recent method for compositional generalization that assumes attributes contribute additively to the classifier's scores rather than to the representation. Our contributions are thus as follows:

- We present LATTICE, a training method that forces a model to imagine the representations of unseen attribute combinations from seen ones, *entirely in an abstract latent space*, so no decoder or pixel reconstruction is needed. Moreover, LATTICE adds no extra parameters.
- We show, theoretically and empirically, that imagination shapes the representation. We prove that exact imagination makes attribute changes parallel on the training combinations, and bound how far this carries over to unseen ones up to an *imagination gap*; on unseen combinations, we measure that LATTICE increases parallelism and reduces this gap.
- We show that LATTICE lets a lightweight backbone outperform a far heavier one under standard training: AIN trained with LATTICE outperforms ED, which dedicates a separate encoder to every attribute and therefore has around 3–6$\times$ more parameters.
- Although the full CRM method underperforms in our experiments, its additive energy loss is effective on its own and complements LATTICE: combining the two yields the best

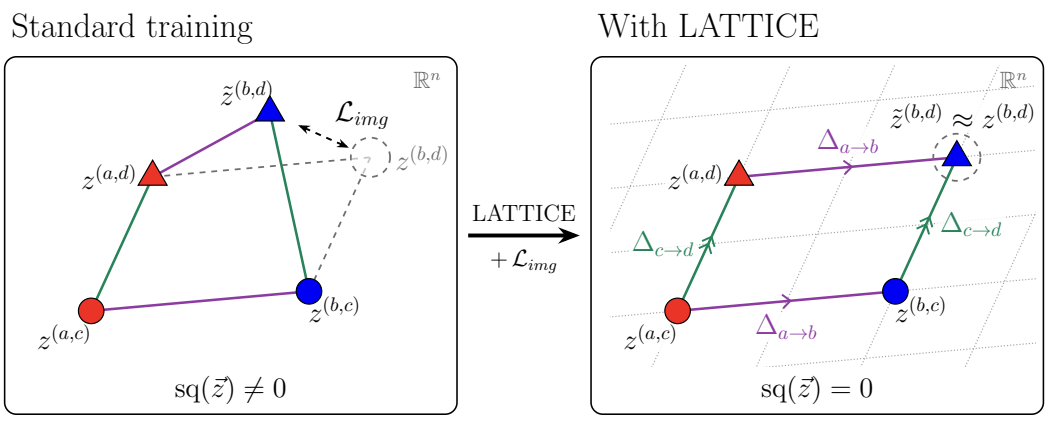

Figure 1: **Imagination shapes the geometry of the representation.** Four training images whose color (a = red, b = blue) and shape (c = circle, d = triangle) form a 2×2 grid are encoded as $z^{(a,c)}, z^{(a,d)}, z^{(b,c)}, z^{(b,d)}$. LATTICE imagines the fourth representation from the other three by analogy, $\tilde{z}^{(b,d)} = z^{(a,d)} + z^{(b,c)} - z^{(a,c)}$, and penalizes its distance to the encoder's actual representation $z^{(b,d)}$ ($\mathcal{L}_{\text{img}}$). **Left:** under standard training, changing the color (purple directions) or the shape (green directions) moves the representation differently depending on the other attribute, so the imagined representation misses the real one (square defect $\mathrm{sq}(\vec{z}) \neq 0$). **Right:** LATTICE reshapes the encoder so that each change is the same displacement, $\Delta_{a\to b}$ or $\Delta_{c\to d}$, wherever it occurs; the four representations form a parallelogram and imagination is exact ($\mathrm{sq}(\vec{z}) = 0$).

average performance on every backbone. On MPI3D, our most realistic benchmark, the best configuration with LATTICE improves on the best configuration without it by 10%.

2 Problem Setup and LATTICE Framework

2.1 Evaluation of Compositional Generalization

Compositional generalization methods have often been evaluated on different datasets and splits of uncontrolled difficulty, which makes them hard to compare. The orthotopic framework (Camposampiero et al., 2025) fixes the difficulty explicitly (Appendix D.1), enabling fair comparisons: the values of each attribute are split into *easy* and *hard* ones, and an image is used for training (in-distribution, ID) if it carries at most c hard values, and for testing (out-of-distribution, OOD) otherwise. With $c = 0$, hard values never appear in training, so the test measures extrapolation to unseen values. We use $c = 1$, the hardest setting for compositional generalization: every hard value is seen in training, but only alongside easy ones, and the test set contains exactly the unseen combinations of two or more hard values. For example, with shape, color and size and hard values □, *green* and *large*, the image (○, *red*, *large*) is a training example, while (○, *green*, *large*) is a test one. Beyond accuracy, we measure the geometry of the representation with the parallelism score (P-score; Bernardi et al., 2020): the cosine between $z^{(a,d)} - z^{(a,c)}$ and $z^{(b,d)} - z^{(b,c)}$, the displacements that the same attribute change produces in two contexts, averaged over all 2×2 grids. It equals 1 when the four representations form a parallelogram (Fig. 1, right).

2.2 LATent Imagination for Compositional Embeddings (LATTICE)

LATTICE trains models on the capability required at test time, systematic recombination, through an auxiliary task in representation space with two components: **(1) Compositional Sampling**, which creates episodes with a missing composition, and **(2) an imagination step**, which predicts the missing representation and is trained jointly with the supervised objective.

1) Compositional Sampling. Each image x has an attribute vector $y = (y_1, \ldots, y_M)$, one value per attribute. An **episode** selects two attributes $p \neq q$, values a, b of p and c, d of q, and fixes all other

attributes, forming a 2×2 cell $y^{(a,c)}, y^{(a,d)}, y^{(b,c)}, y^{(b,d)}$, where the superscript indicates the values assigned to p and q. The first three are observed sources and $y^{(b,d)}$ is the held-out target. Figure 1 illustrates this with colors ■, ■ and shapes △, ○, where the (■,△) target must be predicted from the other combinations.

2) Imagination. Given such an **episode**, LATTICE introduces an auxiliary task: predict the latent representation of the missing composition from the representations of the observed ones. Let $x^{(u,v)}$ be the image associated with $y^{(u,v)}$ and $z^{(u,v)} = f_\theta(x^{(u,v)})$ its encoder latent representation. An imagination rule $\mathcal{I}$ takes the three source latents and returns an imagined target latent:

$$\tilde{z}^{(b,d)} = \mathcal{I}\left(z^{(a,c)}, z^{(a,d)}, z^{(b,c)}\right).$$

We instantiate $\mathcal{I}$ as the parameter-free linear analogy I_{ana} (Mikolov et al., 2013): $\tilde{z}^{(b,d)} = z^{(a,d)} + z^{(b,c)} - z^{(a,c)}$. The imagined latent is trained to match $z^{(b,d)}$, the encoder representation of the held-out composition, jointly with attribute prediction:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \lambda\, \mathcal{L}_{\text{img}},$$

where $\mathcal{L}_{\text{img}} = \|\tilde{z}^{(b,d)} - z^{(b,d)}\|_2^2$ and $\mathcal{L}_{\text{cls}}$ trains the encoder and classifier for attribute prediction. Hyperparameter λ weights the recombination signal. The imagination step applies only during training: at test time the model uses the standard encoder–classifier path. Algorithms 2 and 3 (Appendix D.3) give the full procedure.

3 Theoretical Analysis

The geometric intuition behind LATTICE is that, if the latent space represents two attributes independently, changing one attribute should induce the same displacement regardless of the other. We measure deviations from this geometry by the *square defect* of a tuple $\vec{z} = (z_1, z_2, z_3, z_4)$,

$$\mathsf{sq}(\vec{z}) := z_1 - z_2 - z_3 + z_4 = z_4 - I_{\text{ana}}(z_1, z_2, z_3), \tag{1}$$

which vanishes when the fourth representation is obtained from the other three by the analogy rule. We call the mean representation (see notation) **additive on a set** S if it can be written there as a sum of functions of the individual attribute values. We use this notion to formally justify LATTICE as an effective pressure towards additive latent spaces by highlighting two important theoretical results:

1. **LATTICE induces a parallel geometry in learned representations on the training set:** If the mean representation satisfies the imagination rule on every training cell, it is additive on the training split (Theorem 1); the LATTICE objective encourages this structure.
2. **Generalization to unseen combinations is governed separately by the training objective and architectural constraints:** In the orthotopic split, the transfer bound separates a training defect from an *imagination gap* on unseen combinations (Theorem 2). LATTICE's objective targets the former by encouraging additivity on training combinations. The latter measures whether the encoder places an unseen combination where analogy from seen combinations predicts. Training on seen combinations alone cannot guarantee a small gap; suitable structural conditions on the encoder provide a sufficient way to bound it (Theorem 20).

Notation. Attribute $m \in \{1, \ldots, M\}$ takes values in a finite set Y_m with $|Y_m| \geq 2$, and $Y := Y_1 \times \cdots \times Y_M$ is the set of *configurations*. A configuration $y \in Y$ may be realized by several images, which differ in unlabeled factors; we write $\varphi(y) := \mathbb{E}[f_\theta(x) \mid y] \in \mathbb{R}^n$ for its mean representation, and state all results for φ (the sampled loss adds a within-configuration variance floor, Appendix C.1).

3.1 Induced Additive Latent Structure

Theorem 1 (*Zero square defect implies additive factorization*). Let S be either Y or the $c = 1$ training split of Section 2.1 (more generally, any star-shaped set, Appendix C.2), and assume that for every compositional cell whose four configurations lie in S,

$$I_{\text{ana}}\big(\varphi(y^{(a,c)}), \varphi(y^{(a,d)}), \varphi(y^{(b,c)})\big) = \varphi(y^{(b,d)}). \tag{2}$$

Then there exist maps $\alpha_m : Y_m \to \mathbb{R}^n$, $m = 1, \dots, M$, such that for every configuration $y = (y_1, \dots, y_M) \in S$,

$$\varphi(y_1, \dots, y_M) = \sum_{m=1}^{M} \alpha_m(y_m). \tag{3}$$

This is the discrete characterization of additivity by vanishing mixed differences (Luce & Tukey, 1964). See Appendix E.2 for a discussion on the case of learned analogical rules.

3.2 Generalization to Unseen Combinations in the Orthotopic Split

In the $c = 1$ split (Section 2.1) each value set is partitioned as $Y_m = E_m \sqcup H_m$ into a nonempty set of easy values and a possibly empty set of hard values. We assume at least two attributes have nonempty hard-value sets, so that the test set U below is nonempty. For $y \in Y$ let $T(y) := \{m : y_m \in H_m\}$ be its set of hard coordinates and $j(y) := |T(y)|$; the training and test sets are

$$S := \{y \in Y : |T(y)| \le 1\}, \qquad U := \{y \in Y : |T(y)| \ge 2\},$$

and S splits into the $M + 1$ *strata* $S_0 := \{y : |T(y)| = 0\}$ and $S_1^{(m)} := \{y : T(y) = \{m\}\}$, some of which may be empty. Let $\mathcal{A}$ be the class of additive maps of Theorem 1; empty strata are omitted from the maximum in the following definitions of the *training defect* and *test defect*:

$$\varepsilon_{\text{train}}(\varphi) := \inf_{h \in \mathcal{A}} \max_{\sigma \in \{S_0, S_1^{(1)}, \dots, S_1^{(M)}\}} \mathbb{E}_\sigma \big\|\varphi - h\big\|, \qquad \varepsilon_{\text{test}}(\varphi) := \inf_{h \in \mathcal{A}} \mathbb{E}_U \big\|\varphi - h\big\|, \tag{4}$$

with $\mathbb{E}_\sigma$, $\mathbb{E}_U$ expectations over a uniform configuration of σ and of U.

To quantify the distortion of the representation when encoding multiple hard values, we define the operator Λ as follows: For $y \in U$, draw an *anchor* o by replacing each hard value of y with an independent, uniformly random easy value of the same attribute. The anchor is in S_0, and restoring a single hard value, $o[m \to y_m]$ (o with its m-th coordinate set to y_m, for $m \in T(y)$), gives a configuration in $S_1^{(m)}$. We define

$$\Lambda\varphi(y) := \mathbb{E}_o\Big[\varphi(o) + \sum_{m \in T(y)} \big(\varphi(o[m \to y_m]) - \varphi(o)\big)\Big], \tag{5}$$

the *imagined representation* of y: the representation of the anchor plus the displacement that each hard value of y produces on its own, averaged over anchors (see Proposition 14 for further discussion).

Theorem 2 (*Transfer across the $c = 1$ split*). Let $\bar{\jmath} := \mathbb{E}_U[\,|T(y)|\,] \le M$ be the expected number of hard attributes over the test set. Then the following bound on the test set defect holds:

$$\varepsilon_{\text{test}}(\varphi) \;\le\; \underbrace{\mathbb{E}_U \big\|\varphi - \Lambda\varphi\big\|}_{\text{(I) imagination gap}} \;+\; \underbrace{(2\bar{\jmath} - 1)\, \varepsilon_{\text{train}}(\varphi)}_{\text{(II) training defect}}, \tag{6}$$

and conversely $\big|\, \mathbb{E}_U\|\varphi - h\| - \mathbb{E}_U\|\varphi - \Lambda\varphi\| \,\big| \le (2\bar{\jmath} - 1) \max_\sigma \mathbb{E}_\sigma\|\varphi - h\|$ for every $h \in \mathcal{A}$: the additive model fitted on S is at representation distance (I) from φ on U, up to (II). Proof in Appendix C.5.

Term (I) measures the mean distance between a held-out representation and its analogy-based reconstruction from training configurations. It captures interactions among hard values: with exactly two hard values, the residual is an average of held-out cell defects (Lemma 13). The gap is computable on held-out configurations (Algorithm 1). Term (II) scales $\varepsilon_{\text{train}}$ by $2\bar{\jmath} - 1$; the sampled imagination loss controls this defect under the sampling-coverage conditions of Proposition 12.

Because $\Lambda\varphi$ depends only on representations on S, a small training-support loss or low cardinality and VC-type capacity alone cannot guarantee a small term (I) (Proposition 18). We give a sufficient condition in Theorem 20 to bound term (I) when using attribute-specific encoder architectures, which gives a theoretical justification to the effects observed in our empirical results (Section 5).

4 Experimental Details

Datasets. We evaluate on the six compositional generalization benchmarks of Camposampiero et al. (2025): Cars3D (Reed et al., 2015), dSprites (Matthey et al., 2017), MPI3D (Gondal et al., 2019),

Table 1: Test Accuracy (%) for baseline methods and LATTICE variants using model selection based on ID performance. The second column reports parameter count relative to ResNet-18 on the same dataset; the range spans the six benchmarks, since architectures that instantiate one encoder per attribute grow with the number of attributes. LATTICE and CRM_{loss} are training objectives and add no architectural parameters. Results are averaged over 5 seeds, with standard error of the mean in parentheses. Bold marks the best result per benchmark within each backbone, and * the best result overall, selected by mean accuracy. Shaded rows use LATTICE (ours).

Model	Params	Cars3D	dSprites	MPI3D	CLEVR	I-RAVEN	Shapes3D	Avg
ResNet-18	1.0×	33.12 (0.29)	21.30 (0.39)	43.65 (0.62)	20.40 (2.37)	11.49 (1.63)	84.78 (0.64)	35.79
ResNet-18 + CRM	1.0×	52.95 (0.52)	10.76 (0.51)	30.48 (0.76)	**64.03 (3.68)**	12.54 (1.66)	32.06 (7.63)	33.80
ResNet-18 + CRM_{loss}	1.0×	51.84 (1.31)	16.55 (0.51)	55.65 (1.16)	56.40 (1.72)	19.22 (1.51)	79.64 (2.23)	46.55
ResNet-18 + LATTICE	1.0×	53.99 (0.65)	22.78 (1.26)	43.57 (0.40)	37.89 (3.72)	23.44 (4.01)	87.82 (1.47)	44.91
ResNet-18 + CRM + LATTICE	1.0×	**73.04 (0.47)***	11.12 (0.84)	**59.06 (1.94)**	59.95 (4.07)	27.87 (0.59)	75.49 (2.60)	51.09
ResNet-18 + CRM_{loss} + LATTICE	1.0×	59.36 (1.52)	**24.67 (0.45)**	54.75 (0.54)	60.09 (2.39)	**32.23 (1.90)**	**92.07 (0.83)**	**53.86**
AIN	1.0–1.1×	46.82 (0.89)	55.76 (1.19)	59.55 (2.91)	47.24 (2.41)	59.20 (1.92)	84.07 (2.46)	58.77
AIN + CRM	1.0–1.1×	57.08 (0.48)	27.40 (2.87)	53.26 (0.59)	**69.58 (1.97)***	49.33 (1.87)	51.20 (2.47)	51.31
AIN + CRM_{loss}	1.0–1.1×	61.62 (0.85)	54.06 (0.97)	66.71 (3.37)	67.54 (1.37)	68.65 (2.11)	91.45 (0.32)	68.34
AIN + LATTICE	1.0–1.1×	56.72 (0.51)	**67.25 (1.58)**	57.92 (2.49)	60.50 (0.62)	76.49 (4.18)	**92.89 (1.22)***	68.63
AIN + CRM + LATTICE	1.0–1.1×	**71.14 (0.43)**	49.29 (1.57)	**69.61 (0.85)**	68.70 (0.93)	68.82 (3.80)	80.55 (2.66)	68.02
AIN + CRM_{loss} + LATTICE	1.0–1.1×	60.90 (1.21)	63.26 (2.43)	68.69 (3.36)	60.22 (1.48)	**80.63 (2.88)***	92.70 (1.08)	**71.07**
ED	3.1–6.2×	49.22 (0.47)	**69.99 (0.82)***	68.92 (3.26)	50.33 (2.88)	70.49 (1.37)	90.17 (2.43)	66.52
ED + CRM	3.1–6.2×	61.67 (0.43)	28.68 (4.27)	54.86 (1.25)	67.43 (1.98)	62.09 (2.51)	49.84 (1.54)	54.09
ED + CRM_{loss}	3.1–6.2×	64.83 (0.91)	59.49 (4.26)	71.34 (3.32)	**69.41 (1.22)**	73.78 (2.07)	91.17 (1.34)	71.67
ED + LATTICE	3.1–6.2×	38.00 (0.77)	64.51 (1.26)	64.20 (3.58)	58.52 (3.56)	77.02 (1.75)	89.90 (1.41)	65.36
ED + CRM + LATTICE	3.1–6.2×	**71.88 (0.74)**	58.83 (1.26)	72.30 (0.88)	67.50 (4.14)	70.68 (1.77)	83.49 (2.13)	70.78
ED + CRM_{loss} + LATTICE	3.1–6.2×	61.79 (0.72)	63.24 (0.75)	**78.59 (0.60)***	59.10 (3.67)	**78.77 (2.25)**	**91.21 (1.15)**	**72.12***

CLEVR (Johnson et al., 2017), I-RAVEN (Hu et al., 2021), and Shapes3D (Burgess & Kim, 2018). Five are synthetic; for MPI3D we use the *real* variant, photographs of a robot arm holding simple objects, the only real-world dataset and the one with the most attributes. Appendix D.2 describes the datasets.

Architectures. We use three backbones that span a continuum of architectural bias toward disentanglement, which lets us test whether LATTICE's gains are complementary to that bias. **ResNet-18** (He et al., 2016) is a generic feature learner with no attribute structure; **AIN** (Camposampiero et al., 2025) encourages attribute separation through attribute-specific exclusive blocks followed by shared blocks; and **ED** (Essentially Disentangled; Camposampiero et al., 2025) instantiates one exclusive encoder per attribute with no shared computation. Both were introduced for compositional generalization together with the protocol we adopt, and are the architectural state of the art for this task (Camposampiero et al., 2025). Because ED replicates the backbone once per attribute, its parameter count grows from $3.1\times$ ResNet-18 on the three-attribute benchmarks to $6.2\times$ on MPI3D (Appendix D.5).

Training Methods. We compare standard training, Compositional Risk Minimization (CRM; Mahajan et al., 2025) and LATTICE, each on all three backbones. CRM trains under an additive-energy assumption over attributes and applies a post-hoc bias correction B^* at test time; as discussed in Remark 7, it imposes an analogous additivity commitment, but on the per-attribute energies rather than on the representations. CRM was designed for subpopulation shift, where every attribute group has many examples from which B^* is estimated, an assumption our setting does not meet; we therefore report it in two forms: the full method (**CRM**), and $\textbf{CRM}_{\text{loss}}$, which trains with the same additive-energy group cross-entropy but predicts each attribute independently, isolating the objective from the bias correction.

5 Results

5.1 Overall Results

LATTICE improves lightweight backbones. Table 1 reports the exact-match score, i.e., multi-label accuracy (Camposampiero et al., 2025). LATTICE raises the average accuracy of ResNet-18 from 35.79% to 44.91% ($+25\%$ relative) and of AIN from 58.77% to 68.63% ($+17\%$ relative), improving on five of six benchmarks in both cases; paired tests over the 30 (benchmark, seed) pairs of each backbone confirm both gains ($p < 0.001$; Appendix E.3).

Table 2: Ablations of LATTICE on AIN, as average test accuracy (%) over the six benchmarks. **Left:** the two components — compositional sampling and the imagination loss — removed individually and together, removing both recovering plain AIN. Neither accounts for the effect on its own: their separate contributions sum to less than half the joint gain. **Right:** the imagination weight λ, with architecture, batch construction and schedule held fixed. Accuracy varies by at most 2.1% across $[0.5, 2]$, so the method does not require tuning λ precisely. Per-dataset results in Appendix E.1.

LATTICE components			Imagination Weight	
Model	**Avg Test Acc. (%)**	**Δ (%)**	**λ**	**Avg Test Acc (%)**
			0.5	66.53
AIN + LATTICE	**68.63**	–	0.75	68.49
− sampling	62.80	−5.83	1	66.57
− imagination	59.62	−9.01	1.5	**68.63**
− both (AIN)	58.77	−9.86	2	67.41

On ED, LATTICE alone has little left to add. Adding LATTICE does not significantly change ED's average (66.52% vs 65.36%; $p = 0.97$) and helps on only two of six benchmarks. Its per-attribute encoders are designed to make each block insensitive to the other attributes (Appendix C.8), and it has the highest validation P-score before any compositional objective (0.945, Section 5.3), so the structure LATTICE induces is largely present already.

LATTICE reduces the damage of CRM's bias correction, and is best combined with CRM$_{\text{loss}}$. CRM's extrapolated bias B^* is harmful in our setting: comparing CRM with CRM$_{\text{loss}}$, it costs 12.75% on ResNet-18, 17.03% on AIN and 17.58% on ED, so the additive-energy objective alone outperforms both the full method and the plain baseline. Adding LATTICE to CRM raises accuracy by 16.7 to 17.3% on all three backbones ($p < 0.001$), reducing the damage from B^* to 2.77, 3.05 and 1.34%: LATTICE does not change the data regime that B^* relies on, but largely offsets its cost. Combining LATTICE with CRM$_{\text{loss}}$ gives the best average on every backbone (53.86, 71.07 and 72.12%); on ED it is on par with CRM$_{\text{loss}}$ (72.12% vs 71.67%) but more stable across seeds on four of six benchmarks, most on dSprites (standard error $4.26 \to 0.75$) and MPI3D ($3.32 \to 0.60$).

Results on MPI3D. LATTICE on its own does not help on MPI3D (-0.08, -1.63 and -4.72% on ResNet-18, AIN and ED), but combined with the energy objectives it delivers its largest MPI3D gains ($+28.58\%$ on ResNet-18 + CRM, $+16.35\%$ on AIN + CRM, $+17.44\%$ on ED + CRM), and the best MPI3D result, 78.59%, is ED + CRM$_{\text{loss}}$ + LATTICE, 9.67% above plain ED.

Parameter efficiency. LATTICE and CRM$_{\text{loss}}$ add no parameters, and imagination runs only during training, so inference costs the same as the backbone. ED instead replicates the backbone once per attribute (3.1–6.2× the parameters of ResNet-18). AIN + LATTICE (1.0–1.1×) already outperforms plain ED (68.63% vs 66.52%), and AIN + CRM$_{\text{loss}}$ + LATTICE (71.07%) comes within 1.05% of ED's best configuration (72.12%).

5.2 Ablation Studies

The two components of LATTICE are not independent. Table 2 (left) ablates them on AIN: *compositional sampling*, which draws the four images of each episode as a 2×2 attribute cell (removed by drawing them i.i.d., so that, per Proposition 22, the imagination loss reduces to four times the total variance of the representation, losing compositional content), and the *imagination loss*, which imposes the latent parallelogram constraint on that cell. Removing both recovers plain AIN (58.77%). Removing the sampling costs 5.83% and removing the imagination loss 9.01%, the latter hurting every benchmark, by up to 18.5% on I-RAVEN. Yet on their own they gain only 4.03% (imagination) and 0.85% (sampling) over AIN, 4.88% in total, less than half of the 9.86% they gain together: quadruples are useful only once something consumes their structure, and the parallelogram constraint is meaningful only on batches that actually form parallelograms. Table 2 (right) varies the imagination weight λ with everything else fixed: any $\lambda \in [0.75, 2]$ improves AIN by at least 7.8%, so LATTICE does not depend on tuning λ precisely. The sweep was run only on AIN, on test accuracy since ID validation accuracy saturates for every λ; $\lambda = 1.5$ is used throughout (Appendix D.4).

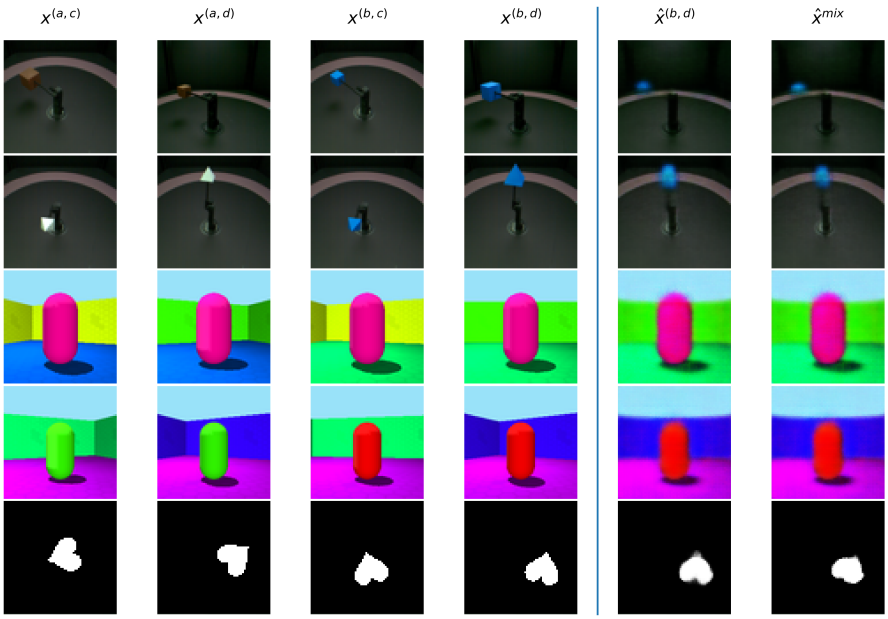

Figure 2: Imagined representations of AIN + LATTICE on held-out test compositions (MPI3D, Shapes3D, dSprites). Columns show the sources $x^{(a,c)}$, $x^{(a,d)}$, $x^{(b,c)}$, the target $x^{(b,d)}$, and the decodings of its true ($\hat{x}^{(b,d)}$) and imagined ($\hat{x}^{mix}$) representations, from a decoder trained *post hoc* on all compositions and used only for visualization (Appendix F.1); more examples in Appendix F.2. Despite some blur, the imagined representation decodes to nearly the same image as the true one and preserves the main factors of the target.

5.3 P-SCORE AND COMPOSITIONAL GENERALIZATION

LATTICE makes the geometry more parallel, also on unseen compositions. The P-score (Section 2.1) is the angular counterpart of the square defect (Appendix C.4). LATTICE raises its average on all three backbones and on both splits (Table 3). Validation nearly saturates (0.97–0.99 on average and at least 0.95 on every benchmark, except 0.85 for ED + LATTICE on Cars3D; Appendix G.1), so the held-out split is where models differ, and there the P-score rises by 0.04 to 0.09: the structure reaches compositions the model never saw. CRM alone moves the P-score by at most 0.06. ED starts highest on validation (0.945) but drops most on test (0.834); adding LATTICE, then CRM, raises its test P-score to 0.872 and 0.910.

LATTICE reduces the imagination gap, the term the theory leaves open. Since the P-score is angular, Table 3 also reports the imagination gap directly, normalized as $\rho_{(\mathrm{I})}$ (term (I) of Theorem 2; Algorithm 1). LATTICE lowers it on every backbone, from 0.634 to 0.497 on ResNet-18, 0.508 to 0.281 on AIN and 0.454 to 0.242 on ED, and adding CRM lowers it further (0.466, 0.251 and 0.212; significantly only on AIN, Appendix E.3). The other term, the square defect on training cells, is not measured directly, but the near-saturated validation P-score suggests that it is small (Theorem 8).

Better geometry predicts, but does not guarantee, accuracy. Within each backbone, validation P-score correlates with test accuracy (Spearman 0.526, 0.555 and 0.391 on ResNet-18, AIN and ED), and $\rho_{(\mathrm{I})}$ does so with the opposite sign (-0.479, -0.532, -0.376; Appendix G.3). A smaller gap is not sufficient, however (Proposition 15): on ED, LATTICE alone halves $\rho_{(\mathrm{I})}$ without improving accuracy, and the gain appears only with the additive-energy objective. Figure 2 shows that imagined and true held-out representations decode to similar images, less so on dSprites.

6 RELATED WORK

Compositional generalization in the generative setting, where the goal is to synthesize an image realizing an unseen combination of factors, spans compositional generation

Table 3: P-score on validation and held-out test representations, and normalized imagination gap $\rho_{(\mathrm{I})}$ (term (I) of Theorem 2; Algorithm 1), averaged over the six benchmarks and 5 seeds. A higher P-score means that changing a factor moves the representation along a more consistent direction; $\rho_{(\mathrm{I})}$ is 0 when every held-out representation equals its imagined one and 1 when imagining is no better than the average anchor. Appendix G.2 breaks $\rho_{(\mathrm{I})}$ down by the number of hard values in the test combination. On every backbone, adding LATTICE, with or without CRM, increases the average P-score on both validation and test and reduces the average $\rho_{(\mathrm{I})}$. Shaded rows use LATTICE (ours).

	ResNet-18			**AIN**			**ED**		
Training method	Val ↑	Test ↑	$\rho_{(\mathrm{I})}$ ↓	Val ↑	Test ↑	$\rho_{(\mathrm{I})}$ ↓	Val ↑	Test ↑	$\rho_{(\mathrm{I})}$ ↓
Base	0.732	0.721	0.634	0.784	0.786	0.508	0.945	0.834	0.454
+ CRM	0.677	0.692	0.638	0.816	0.809	0.440	0.917	0.848	0.370
+ LATTICE	**0.988**	0.810	0.497	**0.991**	0.878	0.281	0.973	0.872	0.242
+ CRM + LATTICE	0.985	**0.840**	**0.466**	0.987	**0.899**	**0.251**	**0.993**	**0.910**	**0.212**

(Nie et al., 2021; Elberg et al., 2025), object-centric and world models (Zhao et al., 2022; Baek et al., 2025), and structured prediction (Okawa et al., 2023). Deep Visual Analogy-Making (Reed et al., 2015), the closest to LATTICE, completes the same vector analogy but decodes it to pixels; LATTICE scores it in latent space, against the encoder's own embedding, and requires no decoder.

Compositional generalization in the discriminative setting. CRM (Mahajan et al., 2025) formulates it through an additive energy decomposition together with a test-time adaptation, and Attribute Invariant Networks (Camposampiero et al., 2025) introduce an architectural bias toward attribute-wise invariance. LATTICE also targets classification, but acts on neither the decision rule nor the architecture: it changes the training procedure itself. Extrapolation from additive structure also underlies additive decoders (Lachapelle et al., 2023), CRM's affine-hull guarantee, and the analyses of Dong & Ma (2023) and Wiedemer et al. (2023) (Appendix C.5). Given recent critiques of claims of systematic compositionality (Woydt et al., 2025), our claims are scoped to the certified $c = 1$ split.

Predicting representations from representations. Joint Embedding Predictive Architectures (LeCun, 2022; Assran et al., 2023; Bardes et al., 2024; Balestriero & LeCun, 2025) predict missing parts of the same input at the representation level, whereas LATTICE predicts the representation of one sample from those of others. Mixup and its latent variants (Zhang et al., 2018; Verma et al., 2019) also combine representations across samples, but interpolate them with their labels as an i.i.d. regularizer, whereas LATTICE extrapolates to the fourth corner of a compositional square, supervised by a real image. Appendix B surveys the wider literature on compositional generalization.

7 Conclusions

Compositional generalization is usually pursued by building structure into the model; LATTICE instead exercises it during training, asking the model to imagine the representation of an attribute combination from three others. Imagination happens entirely in an abstract latent space, with no decoder and no extra parameters to train: the imagined representation is compared with the encoder's own representation of a real image, so the only way to imagine well is to reshape that representation. Our results in the hardest compositional setting suggest three lessons. First, imagination is complementary to architecture: it helps most where the architecture does not already impose the structure, and lets a lightweight backbone outperform one that dedicates a separate encoder to every attribute and is 3–6× larger. Second, Compositional Risk Minimization (CRM) underperforms in this setting, below the plain baseline on every backbone, because of its post-hoc bias correction: its additive-energy objective alone performs far better, on par with LATTICE on ResNet-18 and AIN, and combining the two gives the best average accuracy on every backbone. Third, LATTICE imposes a parallel structure on the representations of seen combinations, and we measure how far it carries over to unseen ones: only partly, since LATTICE brings their representations closer to what imagination predicts, but training on seen combinations alone cannot guarantee the rest, as encoders that agree on them may still differ on unseen ones; architecture and imagination are thus partners rather than substitutes.

AI USE STATEMENT

The core idea and the conceptual framework of this work are the authors' own. We used generative AI tools to help develop the theory, by formulating mathematical claims, providing ingredients for their proofs and assisting in writing them, all of which the authors verified; we also used them to implement methods and to aid in interpreting results (such as producing figures and tables). We have not used generative AI tools to generate synthetic data sets (all datasets were already provided in previous work), propose or refine hypotheses, design or provide feedback for research, assist with translation, clean and reformat datasets, or support qualitative and thematic data analysis. The remaining disclosure categories are not applicable to this work. Additionally, we used generative AI tools to create and modify scientific figures, draft and edit parts of the paper to improve readability, identify relevant literature, and propose a title. We have reviewed all AI-assisted work and checked all proposed literature manually. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only publicly available image benchmarks with ground-truth attribute annotations: five are synthetic and programmatically generated, and one, MPI3D-real, consists of photographs of a robotic arm holding simple objects. None depicts people, and the work involves no human subjects, no personally identifiable information, and no sensitive data. We foresee no direct harmful application of the proposed training strategy.

REPRODUCIBILITY STATEMENT

All six benchmarks are publicly available, and we use the splits of the orthotopic framework (Camposampiero et al., 2025), with the hard values fixed per benchmark (Appendices D.1 and D.2). Architectures, hyperparameters, the choice of λ and checkpoint selection are described in Section 4 and Appendix D.4, and parameter counts and computational resources in Appendices D.5 and D.6. Pseudocode for Compositional Sampling and the LATTICE training step is given in Appendix D.3, and for the normalized imagination gap $\rho_{(\mathrm{I})}$ in Algorithm 1. All results are averaged over 5 seeds with standard errors of the mean, and the statistical tests are detailed in Appendix E.3. Complete proofs of the theoretical claims are given in Appendix C. Our implementation extends that of Camposampiero et al. (2025); the code is provided as anonymized supplementary material and will be released publicly upon publication.

REFERENCES

Panagiotis Alimisis and Christos Diou. Learning robust intervention representations with delta embeddings. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=5d7prMWHNF.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. doi: 10.48550/arXiv.2301.08243. URL https://arxiv.org/abs/2301.08243.

Junyeob Baek, Yi-Fu Wu, Gautam Singh, and Sungjin Ahn. Dreamweaver: Learning compositional world models from pixels. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=e5mTvjXG9u.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025. doi: 10.48550/arXiv.2511.08544. URL https://arxiv.org/abs/2511.08544.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2404.08471.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks, 2018. URL http://arxiv.org/abs/1806.01261.

Silvia Bernardi, Marcus K. Benna, Mattia Rigotti, Jérôme Munuera, Stefano Fusi, and C. Daniel Salzman. The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell*, 2020. doi: 10.1016/j.cell.2020.09.031.

Henry Brighton and Simon Kirby. Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artificial Life*, 12(2):229–242, 2006. doi: 10.1162/106454606776073323.

Chris Burgess and Hyunjik Kim. 3D Shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

Giacomo Camposampiero, Pietro Barbiero, Michael Hersche, Roger Wattenhofer, and Abbas Rahimi. Scalable evaluation and neural models for compositional generalization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=heQsyrMDzm.

Felipe del Rio, Alain Raymond-Saez, Daniel Florea, Rodrigo Toro Icarte, Julio Hurtado, Cristian B. Calderon, and Alvaro Soto. Data distributional properties as inductive bias for systematic generalization. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 25590–25601, June 2025.

Kefan Dong and Tengyu Ma. First steps toward understanding the extrapolation of nonlinear models to unseen domains. In *International Conference on Learning Representations*, 2023.

Rafael Elberg, Felipe del Rio, Mircea Petrache, and Denis Parra. A compressive-expressive communication framework for compositional representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2501.19182.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1):12140, 2017. doi: 10.1038/s41598-017-11873-y.

Karim Farid, Jelena Bratulić, Sudhanshu Mittal, Cordelia Schmid, and Thomas Brox. Latent imagination thinking: Beyond recursive models for reasoning. In *ICLR 2026 the 2nd Workshop on World Models: Understanding, Modelling and Scaling*, 2026a. URL https://openreview.net/forum?id=SYVmt9mFEm.

Karim Farid, Rajat Sahay, Yumna Alnaggar, Simon Schrodi, Volker Fischer, Cordelia Schmid, and Thomas Brox. What drives compositional generalization? the importance of continuous training objectives in visual generative models. In *ICLR 2026 the 2nd Workshop on World Models: Understanding, Modelling and Scaling*, 2026b. URL https://openreview.net/forum?id=k5Kh1XviiM.

Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1–2):3–71, 1988. doi: 10.1016/0010-0277(88)90031-5.

Jingwen Fu, Zhizheng Zhang, Yan Lu, and Nanning Zheng. A general theory for compositional generalization, 2024. URL http://arxiv.org/abs/2405.11743.

Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. 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. URL https://proceedings.neurips.cc/paper/2019/file/d97d404b6119214e4a7018391195240a-Paper.pdf.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

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 (CVPR)*, 2016. doi: 10.1109/CVPR.2016.90.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-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.

Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004. URL https://www.jmlr.org/papers/v5/hoyer04a.html.

Qiyang Hu, Attila Szabó, Tiziano Portenier, Matthias Zwicker, and Paolo Favaro. Disentangling factors of variation by mixing them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3399–3407, 2018. doi: 10.1109/CVPR.2018.00358.

Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. Stratified rule-aware network for abstract visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 1567–1574, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/16248.

Takuya Ito, Tim Klinger, Doug Schultz, John Murray, Michael Cole, and Mattia Rigotti. Compositional generalization through abstract representations in human and artificial neural networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 32225–32239. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/d0241a0fb1fc9be477bdfde5e0da276a-Paper-Conference.pdf.

Jaehyoung Jeong, Hee-Jun Jung, and Kangil Kim. Symmetric space learning for combinatorial generalization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=e8t9F4vX9N.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Zb6c8A-Fghk.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

Frances Y. Kuo, Ian H. Sloan, Grzegorz W. Wasilkowski, and Henryk Woźniakowski. On decompositions of multivariate functions. *Mathematics of Computation*, 79(270):953–966, 2010.

Sébastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien. Additive decoders for latent variables identification and cartesian-product extrapolation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*. PMLR, 2018.

Brenden M. Lake and Marco Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623:115–121, 2023. doi: 10.1038/s41586-023-06668-3.

Daniel Lawson, Adriana Hugessen, Charlotte Cloutier, Glen Berseth, and Khimya Khetarpal. Self-predictive representations for combinatorial generalization in behavioral cloning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=FkeURAdA0h.

Yann LeCun. A path towards autonomous machine intelligence. https://openreview.net/pdf?id=BZ5a1r-kVsf, June 2022. Version 0.9.2, 2022-06-27.

Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. doi: 10.1038/nature14539.

Qiyao Liang, Daoyuan Qian, Liu Ziyin, and Ila Fiete. Compositional generalization via forced rendering of disentangled latents. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2501.18797.

Baihan Lin, Djallel Bouneffouf, and Irina Rish. A survey on compositional generalization in applications, 2023. URL https://arxiv.org/abs/2302.01067.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6781–6792. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/liu21f.html.

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, 2019.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6348–6359. PMLR, 2020a. URL https://proceedings.mlr.press/v119/locatello20a.html.

Francesco Locatello, Michael Tschannen, Stefan Bauer, Gunnar Rätsch, Bernhard Schölkopf, and Olivier Bachem. Disentangling factors of variation using few labels. In *International Conference on Learning Representations (ICLR)*, 2020b.

R. Duncan Luce and John W. Tukey. Simultaneous conjoint measurement: A new type of fundamental measurement. *Journal of Mathematical Psychology*, 1(1):1–27, 1964.

Divyat Mahajan, Mohammad Pezeshki, Charles Arnal, Ioannis Mitliagkas, Kartik Ahuja, and Pascal Vincent. Compositional risk minimization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=axN83wi3NH.

Michael F Mathieu, Junbo Jake Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/ef0917ea498b1665ad6c701057155abe-Paper.pdf.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dSprites: Disentanglement testing sprites dataset. Dataset, 2017. URL https://github.com/google-deepmind/dsprites-dataset.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations*, 2013.

Ishan Misra, Abhinav Gupta, and Martial Hebert. From red wine to red tomato: Composition with context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Milton L. Montero, Casimir J. H. Ludwig, Rui Ponte Costa, Gaurav Malhotra, and Jeffrey S. Bowers. The role of disentanglement in generalisation. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=qbH974jKUVy.

Milton L. Montero, Jeffrey S. Bowers, Rui Ponte Costa, Casimir J. H. Ludwig, and Gaurav Malhotra. Lost in latent space: Disentangled models and the challenge of combinatorial generalisation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://openreview.net/forum?id=7yUxTNWyQGf.

Milton L. Montero, Jeffrey S. Bowers, and Gaurav Malhotra. Successes and limitations of object-centric models at compositional generalisation, 2024. URL https://arxiv.org/abs/2412.18743.

Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Tushar Nagarajan and Kristen Grauman. Attributes as operators: Factorizing unseen attribute-object compositions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

Weili Nie, Arash Vahdat, and Anima Anandkumar. Controllable and compositional generation with latent-space energy-based models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 13497–13510. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/701d804549a4a23d3cae801dac6c2c75-Paper.pdf.

Maya Okawa, Ekdeep Singh Lubana, Robert P. Dick, and Hidenori Tanaka. Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://openreview.net/forum?id=frVo9MzRuU.

Herschel Rabitz and Ömer F. Aliş. General foundations of high-dimensional model representations. *Journal of Mathematical Chemistry*, 25:197–233, 1999.

J. C. Raven. *Progressive Matrices: A Perceptual Test of Intelligence*. H. K. Lewis & Co., London, 1938.

Florian Redhardt, Yassir Akram, and Simon Schug. Scaling can lead to compositional generalization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=hZt0daVIZi.

Scott E. Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/e07413354875be01a996dc560274708e-Abstract.html.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

Lukas Schott, Julius von Kügelgen, Frederik Träuble, Peter Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello, and Wieland Brendel. Visual representation learning does not generalize strongly within the same domain. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=9RUHPlladgh.

Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogerio Feris, Raja Giryes, and Alex M. Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *Advances in Neural Information Processing Systems*, 2018.

Haruki Shirakami and James Henderson. Compositional generalization through gradient search in nonparametric latent space. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RNTWTJe4x6.

Sania Sinha, Tanawan Premsri, and Parisa Kordjamshidi. A survey on compositional learning of AI models: Theoretical and experimental practices. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=BXDxwItNqQ. Survey Certification.

Arnas Uselis, Andrea Dittadi, and Seong Joon Oh. Compositional generalization requires linear, orthogonal representations in vision embedding models. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2602.24264.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6438–6447, 2019. URL https://proceedings.mlr.press/v97/verma19a.html.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 3637–3645, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://openreview.net/forum?id=LqOQ1uJmSx.

Thaddäus Wiedemer, Jack Brady, Alexander Panfilov, Attila Juhos, Matthias Bethge, and Wieland Brendel. Provable compositional generalization for object-centric learning. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=7VPTUWkiDQ.

Tim Woydt, Moritz Willig, Antonia Wüst, Lukas Helff, Wolfgang Stammer, Constantin A. Rothkopf, and Kristian Kersting. Fodor and Pylyshyn's legacy: Still no human-like systematic compositionality in neural networks, 2025. URL https://arxiv.org/abs/2506.01820.

Zhenlin Xu, Marc Niethammer, and Colin Raffel. Compositional generalization in unsupervised compositional representation learning: A study on disentanglement and emergent language. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. RAVEN: A dataset for relational and analogical visual rEasoNing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. URL https://ieeexplore.ieee.org/document/8953364/.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

Linfeng Zhao, Lingzhi Kong, Robin Walters, and Lawson L. S. Wong. Toward compositional generalization in object-oriented world modeling. In *International Conference on Machine Learning (ICML)*, 2022. URL https://proceedings.mlr.press/v162/zhao22b.html.

A LIMITATIONS

LATTICE requires labeled training attributes to construct the compositional quadruples, which restricts it to settings where attribute annotations are available; extending it to cases where the factors must themselves be discovered is left to future work. Like most work on compositional generalization (Camposampiero et al., 2025; Mahajan et al., 2025; Schott et al., 2022; Montero et al., 2021), we evaluate mostly on synthetic benchmarks, where attributes are fully labeled and every 2×2 training cell is available (Appendix D.2); these controlled settings remain unsolved, and extending LATTICE to natural images, where many attribute combinations are never observed, is left to future work. The analogy rule also commits to factors combining additively in representation space, a constraint that may be less appropriate for attributes whose effects interact strongly. Our theory guarantees additivity only on the training combinations: the transfer bound of Section 3.2 carries it to held-out combinations up to the imagination gap (term (I)), which training on seen combinations alone cannot guarantee to be small. Reducing this term further is a matter of architectural bias, outside the scope of this paper; we measure it rather than prove it small. The two terms of the bound are not reported on a common scale: we measure the imagination gap directly, but only estimate the training term indirectly from the validation P-score. Finally, ED delimits the regime where the method helps on its own: when an architecture already yields a near-parallel geometry by design on the training support, LATTICE has little left to impose there, and on ED it improves accuracy only in combination with the additive-energy objectives; ED's gains come at the cost of a parameter count that ResNet-18 and AIN avoid.

B Extended Related Work

Section 6 positions LATTICE against the methods it is most directly compared with. Here we survey the wider literature.

Compositional generalization. Compositional generalization is a longstanding challenge in machine learning, studied across a range of modalities, tasks and architectural settings (Lake & Baroni, 2018; 2023; Camposampiero et al., 2025). Whether disentanglement facilitates it by separating the underlying factors of variation remains contested: some work finds that more abstract or disentangled representations support better compositional behavior (Montero et al., 2021; Xu et al., 2022; Ito et al., 2022), while others argue that disentanglement alone is insufficient and does not reliably yield strong combinatorial generalization (Montero et al., 2022; Liang et al., 2025). More broadly, suitable inductive biases appear necessary for compositional generalization to emerge (Battaglia et al., 2018; Wiedemer et al., 2023; Fu et al., 2024). Prior work has also pointed out that the continuous nature of attributes has an effect on compositional generalization (Farid et al., 2026b). Recent efforts have therefore explored object-centric representations (Wiedemer et al., 2024; Montero et al., 2024; Zhao et al., 2022), increasing data diversity and scale (del Rio et al., 2025; Redhardt et al., 2025) or leveraging temporal information in video as mechanisms to encourage more systematic generalization (Baek et al., 2025). They are not sufficient, however: models equipped with them may still rely on shortcuts or memorization (Schott et al., 2022; Montero et al., 2022; 2024). We refer the reader to broader surveys (Lin et al., 2023; Sinha et al., 2024) for further reading.

Training methods for compositional generalization. Beyond the generative work discussed in Section 6, recent approaches have addressed reasoning (Shirakami & Henderson, 2026; Farid et al., 2026a) and reinforcement learning (Lawson et al., 2026). Another line aims to reduce explicit supervision by inferring compositional structure directly from data (Xu et al., 2022; Locatello et al., 2020b;a; Hu et al., 2018). In the discriminative setting, robustness to distribution shift has been widely studied (Sagawa et al., 2020; Liu et al., 2021; Kirichenko et al., 2023), but only a few methods explicitly target compositional generalization there; we discuss them in Section 6.

Structure on attribute directions. More broadly, LATTICE relates to approaches that impose structure on attribute directions in representation space. Alimisis & Diou (2026) constrain attribute-related directions using contrastive losses, whereas LATTICE does so through an imagination objective. Similarly, Jeong et al. (2026) use the training distribution to approximate the test distribution through symmetry, while LATTICE does so by exploiting attribute decompositions in latent space. Prior work has also used latent imagination for long-horizon planning (Hafner et al., 2020), rather than for systematic generalization.

Related settings. Compositional zero-shot learning composes learned primitives to recognize unseen attribute–object pairs in natural images (Misra et al., 2017; Nagarajan & Grauman, 2018; Naeem et al., 2021), where most pairs are unobserved. Feature hallucination (Hariharan & Girshick, 2017; Schwartz et al., 2018) transfers displacements by analogy, as LATTICE does, but to synthesize features for new classes rather than to constrain the encoder.

C Theoretical Results and Proofs

C.1 Preliminaries

Throughout, $Y = Y_1 \times \cdots \times Y_M$ is the configuration grid, and $\varphi(y) := \mathbb{E}[f_\theta(x) \mid y]$ the mean representation of the images realizing $y \in Y$, as in Section 2; when y is realized by a single image $x(y)$, $\varphi(y) = f_\theta(x(y))$. We extend the square defect of (1) to any map g defined on configurations,

$$\mathsf{sq}\, g(y_1, y_2, y_3, y_4) := g(y_1) - g(y_2) - g(y_3) + g(y_4), \tag{7}$$

and for $g = \mathrm{Id}$ on $\mathbb{R}^n$ we simply write $\mathsf{sq}(z_1, z_2, z_3, z_4) := z_1 - z_2 - z_3 + z_4$. Recall te definition of the analogic imagination rule that "completes the parallelogram" with consecutive vertices z_2, z_1, z_3 by producing its fourth vertex:

$$I_{ana}(z_1, z_2, z_3) := z_2 + z_3 - z_1. \tag{8}$$

Thus I_{ana} is characterized as the only map satisfying $\mathsf{sq}(z_1, z_2, z_3, I_{\mathrm{ana}}(z_1, z_2, z_3)) = 0$ for all $z_1, z_2, z_3 \in \mathbb{R}^n$.

Two elementary facts about sq are used repeatedly: it is linear in g, and it vanishes identically on the additive class $\mathcal{A} = \{h : \ h(y) = \sum_m \alpha_m(y_m)\}$, since on a cell each α_m appears twice with sign $+$ and twice with sign $-$.

Recall that a **compositional cell** in the space of generators is an ordered tuple $(y^{(a,c)}, y^{(a,d)}, y^{(b,c)}, y^{(b,d)})$ which can be brought to the following form, up to reordering of the factors (where r is a context string including all but 2 of the factors):

$$y^{(u,v)} = (u, v, r), \qquad \text{for} \qquad (u, v) \in \{(a, c), (a, d), (b, c), (b, d)\}, \tag{9}$$

with $a \neq b$ and $c \neq d$. We will also use *degenerate* tuples of the same form with $a = b$ or $c = d$; for these two of the four configurations coincide pairwise and sq vanishes trivially, so hypotheses stated "on every cell" hold on degenerate tuples for free.

Star-shaped sets. For $o, y \in Y$ let $\mathrm{Box}_o(y) := \{y' \in Y : \ y'_m \in \{y_m, o_m\} \text{ for all } m\}$ be the sub-box spanned by y and o. A set $S \subseteq Y$ is *star-shaped about* o if $y \in S$ implies $\mathrm{Box}_o(y) \subseteq S$; in particular $o \in S$ as soon as $S \neq \emptyset$. Every cell all of whose corners lie in $\mathrm{Box}_o(y)$ is a cell of S, which is the only property of star-shapedness used below.

Several observations per configuration. If a configuration y is realized by several observations, let $\varphi(y) := \mathbb{E}[f_\theta(x) \mid y]$ and write the encoder output of a random observation of y as $\varphi(y) + \xi$ with $\mathbb{E}[\xi \mid y] = 0$. When the four members of an episode are drawn independently given their configurations, the imagination loss on a cell equals

$$\mathbb{E}\big\|\mathsf{sq}(z_1, z_2, z_3, z_4)\big\|^2 = \big\|\mathsf{sq}\,\varphi(y_1, y_2, y_3, y_4)\big\|^2 + \sum_{i=1}^{4} \mathbb{E}\|\xi_i\|^2,$$

since the cross terms vanish by independence and centering. The sampled loss therefore upper-bounds the square defect of the conditional-mean encoder, and every statement below applies to φ with $\epsilon := \sup_{\text{cells}} \|\mathsf{sq}\,\varphi\| \leq \sup_{\text{cells}} \sqrt{\mathbb{E}\|\mathsf{sq}(z)\|^2}$; the within-configuration variance is an additive floor on the loss that carries no compositional information. In particular, when images of the same configuration differ, a zero imagination loss is unattainable, and the hypotheses of our results concern the square defect of φ, not the loss.

C.2 Case of Zero Square Defect

We show that if an encoding has zero square defect on the cells of a star-shaped set, then it is additively separable on that set. We prove the two-factor case first and then extend it to M factors by induction.

Notation: *As in Appendix C.1,* $\varphi(y) := \mathbb{E}[f_\theta(x) \mid y]$ *is the mean representation of configuration* $y = (y_1, \ldots, y_M) \in Y$.

Proposition 3 (Exact separability, two-factor case). Let V and W be finite sets with $|V|, |W| \geq 2$. Let $\varphi : V \times W \to \mathbb{R}^n$ be any map satisfying, for all $s_1, s_2 \in V$ and $t_1, t_2 \in W$:

$$\varphi(s_2, t_2) = \varphi(s_1, t_2) - \varphi(s_1, t_1) + \varphi(s_2, t_1). \tag{$\star$}$$

Then φ is additively separable: there exist $\alpha : V \to \mathbb{R}^n$ and $\beta : W \to \mathbb{R}^n$ such that

$$\varphi(s, t) = \alpha(s) + \beta(t) \quad \text{for all } (s, t) \in V \times W.$$

Proof. Fix arbitrary reference elements $s_0 \in V$, $t_0 \in W$. Define

$$\alpha(s) := \varphi(s, t_0), \qquad \beta(t) := \varphi(s_0, t) - \varphi(s_0, t_0).$$

For any (s, t), apply ($\star$) with $s_1 = s_0$, $s_2 = s$, $t_1 = t_0$, $t_2 = t$:

$$\varphi(s, t) = \varphi(s_0, t) - \varphi(s_0, t_0) + \varphi(s, t_0) = \beta(t) + \alpha(s). \qquad \square$$

Definition 1. A set of configurations $S \subseteq Y$ is *star-shaped about* $o \in S$ if, whenever $y \in S$, every configuration whose m-th coordinate lies in $\{y_m, o_m\}$ for all m also lies in S: S contains the sub-box spanned by y and o.

The full grid Y is star-shaped about any of its points, and so is the $c = 1$ training split of the Orthotopic framework about any configuration made of easy values (Section 3.2). In particular, the below theorem extends the statement of Theorem 1.

Theorem 4 (Full separability on a star-shaped set; proof of Theorem 1). Let $S \subseteq Y$ be star-shaped about $o = (o_1, \ldots, o_M) \in S$, and suppose the analogic imagination module I_{ana} achieves exact square completion on every cell whose four configurations lie in S. Define

$$\begin{aligned} \alpha_1(y_1) &:= \varphi(y_1, o_2, \ldots, o_M), \\ \alpha_m(y_m) &:= \varphi(o_1, \ldots, o_{m-1}, y_m, o_{m+1}, \ldots, o_M) - \varphi(o), \quad m \geq 2. \end{aligned}$$

Then

$$\varphi(y_1, \ldots, y_M) = \sum_{m=1}^{M} \alpha_m(y_m) \qquad \text{for every } y \in S.$$

Taking $S = Y$ recovers additivity on the whole grid.

Proof. Fix $y \in S$. Every configuration appearing in the definition of the α_m and in the argument below has its m-th coordinate in $\{y_m, o_m\}$ for every m, i.e., lies in $\mathrm{Box}_o(y) \subseteq S$; we check this as we go, so that every cell invoked is a cell of S and has zero defect by hypothesis (degenerate tuples having zero defect trivially).

We show by induction on M that $\varphi(y_1, \ldots, y_M) = \sum_{m=1}^{M} \alpha_m(y_m)$.

Base case $M = 2$. This is Proposition 3 applied to φ restricted to $\{y_1, o_1\} \times \{y_2, o_2\} \subseteq S$, with $s_0 = o_1$, $t_0 = o_2$.

Inductive step. Assume the result holds for $M - 1$ factors. Decompose the residual as:

$$\begin{aligned} \varphi(y_1, \ldots, y_M) - \sum_{m=1}^{M} \alpha_m(y_m) = &\underbrace{\Big[\varphi(y_1, \ldots, y_M) - \varphi(y_1, \ldots, y_{M-1}, o_M) - \alpha_M(y_M)\Big]}_{=: P} \\ &+ \underbrace{\Big[\varphi(y_1, \ldots, y_{M-1}, o_M) - \sum_{m=1}^{M-1} \alpha_m(y_m)\Big]}_{=: Q}. \end{aligned}$$

Vanishing of Q. The function $g(y_1, \ldots, y_{M-1}) := \varphi(y_1, \ldots, y_{M-1}, o_M)$ is defined on the box $\prod_{m<M}\{y_m, o_m\}$, all of whose points lie in $\mathrm{Box}_o(y)$; it satisfies the zero square defect condition on every cell of that box (fixing $y_M = o_M$ yields a valid context), and its α's coincide with $\alpha_1, \ldots, \alpha_{M-1}$. By the inductive hypothesis, $Q = 0$.

Vanishing of P. Recalling $\alpha_M(y_M) = \varphi(o_1, \ldots, o_{M-1}, y_M) - \varphi(o)$, write

$$P = \varphi(y_1, \ldots, y_M) - \varphi(y_1, \ldots, y_{M-1}, o_M) - \varphi(o_1, \ldots, o_{M-1}, y_M) + \varphi(o).$$

Define $A_k := \varphi(y_1, \ldots, y_k, o_{k+1}, \ldots, o_{M-1}, y_M)$ and $B_k := \varphi(y_1, \ldots, y_k, o_{k+1}, \ldots, o_{M-1}, o_M)$, with $A_0 := \varphi(o_1, \ldots, o_{M-1}, y_M)$ and $B_0 := \varphi(o)$. Then $P = (A_{M-1} - B_{M-1}) - (A_0 - B_0)$, which telescopes as

$$P = \sum_{k=1}^{M-1} \big[(A_k - B_k) - (A_{k-1} - B_{k-1})\big] =: \sum_{k=1}^{M-1} E_k.$$

Each E_k is precisely the square defect of φ on the tuple for the pair (k, M) with context $(y_1, \ldots, y_{k-1}, o_{k+1}, \ldots, o_{M-1})$, values $\{y_k, o_k\}$ for attribute k and $\{y_M, o_M\}$ for attribute M. Its four corners lie in $\mathrm{Box}_o(y) \subseteq S$, so $E_k = 0$ by assumption (trivially if $y_k = o_k$ or $y_M = o_M$). Therefore $P = 0$.

Conclusion. Both P and Q vanish, giving $\varphi(y_1, \ldots, y_M) = \sum_{m=1}^{M} \alpha_m(y_m)$. □

Remark 1 (Why star-shapedness). The hypothesis of Theorem 4 is stated on cells of S only, so it cannot conclude anything outside S: any ψ with $\psi = \varphi$ on S satisfies the same hypothesis. Some connectivity condition on S is nevertheless needed even for the conclusion on S: a set containing no cell at all satisfies the hypothesis vacuously. For example, in a 3×3 grid the "hexagon" $\{(1,1), (1,2), (2,2), (2,3), (3,3), (3,1)\}$ contains no cell, while the restrictions of additive functions to it span only 5 of its 6 dimensions, so non-additive encoders satisfy the hypothesis. Star-shapedness is the simplest condition under which the telescoping above stays inside S; both the full grid and the $c = 1$ training split satisfy it.

C.3 Case of Small Square Defect

Notation: *As in Appendix C.1, $\varphi(y) := \mathbb{E}[f_\theta(x) \mid y]$ is the mean representation of configuration $y = (y_1, \ldots, y_M) \in Y$.*

Proposition 5 (Approximate separability, two-factor case). Let V and W be finite sets with $|V|, |W| \geq 2$. Let $\varphi : V \times W \to \mathbb{R}^n$ satisfy, for all $s_1, s_2 \in V$ and $t_1, t_2 \in W$:

$$\big\|\mathsf{sq}\,\varphi\big((s_1,t_1),(s_1,t_2),(s_2,t_1),(s_2,t_2)\big)\big\| \leq \varepsilon. \tag{$\star_\varepsilon$}$$

Fix reference elements $s_0 \in V$, $t_0 \in W$ and define

$$\alpha(s) := \varphi(s, t_0), \qquad \beta(t) := \varphi(s_0, t) - \varphi(s_0, t_0).$$

Then

$$\big\|\varphi(s,t) - \alpha(s) - \beta(t)\big\| \leq \varepsilon \qquad \text{for all } (s,t) \in V \times W.$$

Proof. The proof is identical in structure to the exact case. Computing directly:

$$\varphi(s,t) - \alpha(s) - \beta(t) = \varphi(s,t) - \varphi(s,t_0) - \varphi(s_0,t) + \varphi(s_0,t_0) = \mathsf{sq}\,\varphi\big((s_0,t_0),(s_0,t),(s,t_0),(s,t)\big),$$

whose norm is at most ε by assumption ($\star_\varepsilon$). □

Corollary 1 (Approximate full separability on a star-shaped set). Let $S \subseteq Y$ be star-shaped about $o \in S$ and suppose every cell whose four configurations lie in S has square defect bounded by ε. With α_m defined from the reference point o as in Theorem 4,

$$\left\|\varphi(y_1, \ldots, y_M) - \sum_{m=1}^{M} \alpha_m(y_m)\right\| \leq \binom{M}{2}\varepsilon \qquad \text{for every } y \in S.$$

In particular, there is a *single* $h = \sum_m \alpha_m \in \mathcal{A}$ within $\binom{M}{2}\varepsilon$ of φ uniformly on S.

Proof. By induction on M, following the proof of Theorem 4 verbatim; as there, every configuration and every cell that appears lies in $\mathrm{Box}_o(y) \subseteq S$.

Base case $M = 2$. The approximate proposition gives error $\leq \varepsilon = \binom{2}{2}\varepsilon$.

Inductive step. Assume the bound holds for $M-1$ factors with constant $\binom{M-1}{2}$. Decompose the residual as:

$$\varphi(y_1,\dots,y_M) - \sum_{m=1}^{M} \alpha_m(y_m) = \underbrace{\Big[\varphi(y_1,\dots,y_M) - \varphi(y_1,\dots,y_{M-1},o_M) - \alpha_M(y_M)\Big]}_{=:\,P} + \underbrace{\Big[\varphi(y_1,\dots,y_{M-1},o_M) - \sum_{m=1}^{M-1} \alpha_m(y_m)\Big]}_{=:\,Q}.$$

Bounding Q. The function $g(y_1,\dots,y_{M-1}) := \varphi(y_1,\dots,y_{M-1},o_M)$ has cell defect $\leq \varepsilon$ on the box $\prod_{m<M}\{y_m, o_m\}$ (fixing $y_M = o_M$ yields a valid context). By the inductive hypothesis, $\|Q\| \leq \binom{M-1}{2}\varepsilon$.

Bounding P. With A_k, B_k as in the proof of Theorem 4, $P = \sum_{k=1}^{M-1} E_k$ where each E_k is the square defect of φ on a tuple with corners in $\mathrm{Box}_o(y)$, hence $\|E_k\| \leq \varepsilon$ and $\|P\| \leq (M-1)\varepsilon$.

Conclusion.

$$\left\|\varphi(y_1,\dots,y_M) - \sum_{m=1}^{M} \alpha_m(y_m)\right\| \;\leq\; \binom{M-1}{2}\varepsilon + (M-1)\varepsilon \;=\; \binom{M}{2}\varepsilon. \qquad \square$$

C.4 Square Defect vs. P-score

In this subsection $f_\theta(x^{(u,v)})$ stands for the mean representation $\varphi(y^{(u,v)})$ of the configuration $y^{(u,v)}$, so that the P-score and the square defect below are those of the conditional-mean encoder; when each configuration is realized by a single image, the two coincide. As before, we will consider values a, b for attribute p and values c, d for attribute $q \neq p$ and let the values of remaining attributes be fixed and denoted by r and use notation as before. The P-score (parallelism score) of the tuple of the form (9) with $x^{(u,v)}$ denoting the image associated to attribute values $y^{(u,v)}$, is

$$\mathsf{ps} f_\theta(y^{(a,c)}, y^{(a,d)}, y^{(b,c)}, y^{(b,d)}) := \frac{f_\theta(x^{(a,d)}) - f_\theta(x^{(a,c)})}{\|f_\theta(x^{(a,d)}) - f_\theta(x^{(a,c)})\|} \cdot \frac{f_\theta(x^{(b,d)}) - f_\theta(x^{(b,c)})}{\|f_\theta(x^{(b,d)}) - f_\theta(x^{(b,c)})\|},$$

and the global P-score is the average of $\mathsf{ps} f_\theta$ over all choices of attributes p, q and of values a, b of p, values c, d of q and contexts r.

Proposition 6 (Square defect control on P-score). Let $f_\theta : X \to \mathbb{R}^n$. For each compositional tuple $(y^{(a,c)}, y^{(a,d)}, y^{(b,c)}, y^{(b,d)})$ as in (9), denote by $x^{(u,v)}$ the image associated to attribute values $y^{(u,v)}$ and denote the difference vectors

$$\ell_1 := f_\theta(x^{(a,d)}) - f_\theta(x^{(a,c)}), \qquad \ell_2 := f_\theta(x^{(b,d)}) - f_\theta(x^{(b,c)}).$$

Let $\overline{\mathsf{sq}^2} := \mathbb{E}[\|\mathsf{sq}\, f_\theta\|^2]$ and $\overline{\mathsf{ps}} := \mathbb{E}[\mathsf{ps}\, f_\theta]$ denote the averages over all such compositional tuples (i.e. over all choices of attribute pair p, q, and of values a, b for p and c, d for q, and contexts r, as in the paragraph preceding (9)).

Then there holds

$$\overline{\mathsf{sq}^2} = \mathbb{E}\big[(\|\ell_1\| - \|\ell_2\|)^2\big] + 2\,\mathbb{E}\big[\|\ell_1\|\|\ell_2\|\,(1 - \mathsf{ps}\, f_\theta)\big]. \tag{10}$$

Furthermore if we assume the encoding is *non-degenerate* in the sense that $\|\ell_1\|, \|\ell_2\| \geq \delta > 0$ for all compositional squares, then:

$$\overline{\mathsf{ps}} \;\geq\; 1 - \frac{\overline{\mathsf{sq}^2}}{2\delta^2}.$$

Proof. We first establish a pointwise identity. Expanding directly:

$$\|\mathsf{sq}\, f_\theta\|^2 = \|\ell_1 - \ell_2\|^2 = \|\ell_1\|^2 - 2\,\ell_1 \cdot \ell_2 + \|\ell_2\|^2.$$

Since $\ell_1 \cdot \ell_2 = \|\ell_1\|\|\ell_2\|\,\mathsf{ps}\, f_\theta$ by definition of the P-score, and since $\|\ell_1\|^2 + \|\ell_2\|^2 = (\|\ell_1\| - \|\ell_2\|)^2 + 2\|\ell_1\|\|\ell_2\|$, we obtain

$$\|\mathsf{sq}\, f_\theta\|^2 = (\|\ell_1\| - \|\ell_2\|)^2 + 2\|\ell_1\|\|\ell_2\| - 2\|\ell_1\|\|\ell_2\|\,\mathsf{ps}\, f_\theta = (\|\ell_1\| - \|\ell_2\|)^2 + 2\|\ell_1\|\|\ell_2\|(1 - \mathsf{ps}\, f_\theta).$$

Since both terms on the right are non-negative (as $\mathsf{ps} f_\theta \le 1$), averaging over all compositional tuples gives (10).

Next, from (10), since $(\|\ell_1\| - \|\ell_2\|)^2 \ge 0$:

$$\overline{\mathsf{sq}^2} \;\ge\; 2\,\mathbb{E}\big[\|\ell_1\|\|\ell_2\|(1 - \mathsf{ps}\, f_\theta)\big] \;\ge\; 2\delta^2\,\mathbb{E}\big[1 - \mathsf{ps}\, f_\theta\big] = 2\delta^2(1 - \overline{\mathsf{ps}}),$$

where the second inequality uses $\|\ell_1\|, \|\ell_2\| \ge \delta$ pointwise. Rearranging yields $\overline{\mathsf{ps}} \ge 1 - \overline{\mathsf{sq}^2}/(2\delta^2)$. □

Proposition 7 (P-score control on square defect). For a compositional tuple $(y^{(a,c)}, y^{(a,d)}, y^{(b,c)}, y^{(b,d)})$ as in (9) and denoting by $x^{(u,v)}$ the image associated to $y^{(u,v)}$, define the two pairs of opposite-side vectors

$$\ell_1 := f_\theta(x^{(a,d)}) - f_\theta(x^{(a,c)}), \qquad \ell_2 := f_\theta(x^{(b,d)}) - f_\theta(x^{(b,c)}),$$
$$\ell_3 := f_\theta(x^{(a,d)}) - f_\theta(x^{(b,d)}), \qquad \ell_4 := f_\theta(x^{(a,c)}) - f_\theta(x^{(b,c)}),$$

and the associated P-scores

$$\mathsf{ps}_A := \frac{\ell_1 \cdot \ell_2}{\|\ell_1\|\|\ell_2\|}, \qquad \mathsf{ps}_B := \frac{\ell_3 \cdot \ell_4}{\|\ell_3\|\|\ell_4\|}.$$

Let $\vartheta := \angle(\ell_1, \ell_3)$ be the angle between the two adjacent sides, and assume $\|f_\theta(x)\| \le C$ for all x, so that $\|\ell_1\|, \|\ell_2\|, \|\ell_3\|, \|\ell_4\| \le 2C$. Then:

$$\|\mathsf{sq}\, f_\theta\|^2 \;\le\; \frac{24C^2}{\sin^2\vartheta}\Big[(1 - \mathsf{ps}_A) + (1 - \mathsf{ps}_B)\Big]. \tag{11}$$

In particular, under the *non-degeneracy* assumption $\sin\vartheta \ge \delta > 0$ uniformly over all compositional tuples, taking averages and using the symmetry $\mathbb{E}[1 - \mathsf{ps}_A] = \mathbb{E}[1 - \mathsf{ps}_B] = 1 - \overline{\mathsf{ps}}$ yields

$$\overline{\mathsf{sq}^2} \;\le\; \frac{48C^2}{\delta^2}\,(1 - \overline{\mathsf{ps}}\,). \tag{12}$$

Proof. The geometric content is the following: in a quadrilateral, if *both* pairs of opposite sides are nearly parallel, then it is nearly a parallelogram – provided the quadrilateral is non-degenerate (i.e. the adjacent sides are not collinear). We make this quantitative.

Step 1: perpendicular components. Since $\mathsf{sq}\, f_\theta = \ell_2 - \ell_1$ (resp. $= \ell_4 - \ell_3$), the component of sq perpendicular to ℓ_1 equals $-\ell_{2\perp}$, the component of ℓ_2 perpendicular to ℓ_1. Hence

$$\|\mathsf{sq}_\perp^{\ell_1}\|^2 := \|\mathsf{sq} - (\mathsf{sq} \cdot \hat\ell_1)\hat\ell_1\|^2 = \|\ell_2\|^2(1 - \mathsf{ps}_A^2) \le 4C^2 \cdot 2(1 - \mathsf{ps}_A) = 8C^2\varepsilon_A,$$

where $\varepsilon_A := 1 - \mathsf{ps}_A$ and we used $1 - \mathsf{ps}_A^2 \le 2(1 - \mathsf{ps}_A)$. Identically, the component perpendicular to ℓ_3 satisfies

$$\|\mathsf{sq}_\perp^{\ell_3}\|^2 = \|\ell_4\|^2(1 - \mathsf{ps}_B^2) \le 8C^2\varepsilon_B, \qquad \varepsilon_B := 1 - \mathsf{ps}_B.$$

Step 2: decomposition relative to $\mathrm{span}(\ell_1, \ell_3)$**.** Write $\mathsf{sq} = \mathsf{sq}_{2D} + \mathsf{sq}^\perp$ where $\mathsf{sq}_{2D} \in \mathrm{span}(\ell_1, \ell_3)$ and $\mathsf{sq}^\perp \perp \mathrm{span}(\ell_1, \ell_3)$. Let $\{\hat\ell_1, \hat e\}$ be an orthonormal basis of $\mathrm{span}(\ell_1, \ell_3)$ with $\hat e \perp \hat\ell_1$, so that $\hat\ell_3 = \cos\vartheta\,\hat\ell_1 + \sin\vartheta\,\hat e$. Write $\mathsf{sq}_{2D} = t_1\hat\ell_1 + t_2\hat e$. Since the projections onto $\mathrm{span}(\ell_1, \ell_3)^\perp$ and onto the in-plane orthogonal complements are orthogonal to each other, we have

$$\|\mathsf{sq}_\perp^{\ell_1}\|^2 = t_2^2 + \|\mathsf{sq}^\perp\|^2,$$

so $\|\mathsf{sq}\|^2 = t_1^2 + t_2^2 + \|\mathsf{sq}^\perp\|^2 = t_1^2 + \|\mathsf{sq}_\perp^{\ell_1}\|^2$. Hence, using Step 1,

$$\|\mathsf{sq}\|^2 = t_1^2 + \|\mathsf{sq}_\perp^{\ell_1}\|^2 \le t_1^2 + 8C^2\varepsilon_A. \tag{13}$$

Step 3: bounding the parallel component t_1. A direct computation gives $\|\mathsf{sq}_{2D,\perp}^{\ell_3}\|^2 = (t_1 \sin\vartheta - t_2\cos\vartheta)^2$, and since $\mathsf{sq}^{\perp} \perp \ell_3$ we have $\|\mathsf{sq}_{\perp}^{\ell_3}\|^2 = \|\mathsf{sq}_{2D,\perp}^{\ell_3}\|^2 + \|\mathsf{sq}^{\perp}\|^2 \geq (t_1\sin\vartheta - t_2\cos\vartheta)^2$, so by Step 1:

$$|t_1\sin\vartheta - t_2\cos\vartheta| \leq \sqrt{\|\mathsf{sq}_{\perp}^{\ell_3}\|^2} \leq 2C\sqrt{2\varepsilon_B}.$$

Combining with $|t_2| \leq \sqrt{\|\mathsf{sq}_{\perp}^{\ell_1}\|^2} \leq 2C\sqrt{2\varepsilon_A}$:

$$\sin\vartheta\,|t_1| \leq |t_1\sin\vartheta - t_2\cos\vartheta| + |\cos\vartheta|\,|t_2| \leq 2C\sqrt{2\varepsilon_B} + 2C\sqrt{2\varepsilon_A} = 2C\sqrt{2}\,(\sqrt{\varepsilon_A} + \sqrt{\varepsilon_B}),$$

hence

$$t_1^2 \leq \frac{8C^2(\sqrt{\varepsilon_A} + \sqrt{\varepsilon_B})^2}{\sin^2\vartheta} \leq \frac{16C^2(\varepsilon_A + \varepsilon_B)}{\sin^2\vartheta},$$

where we used $(\sqrt{a} + \sqrt{b})^2 \leq 2(a+b)$.

Step 4: conclusion. Substituting into (13) and using $\sin\vartheta \leq 1$:

$$\|\mathsf{sq}\|^2 \leq \frac{16C^2(\varepsilon_A + \varepsilon_B)}{\sin^2\vartheta} + 8C^2\varepsilon_A \leq \frac{16C^2(\varepsilon_A + \varepsilon_B)}{\sin^2\vartheta} + \frac{8C^2(\varepsilon_A + \varepsilon_B)}{\sin^2\vartheta} = \frac{24C^2(\varepsilon_A + \varepsilon_B)}{\sin^2\vartheta},$$

which is (11). Under $\sin\vartheta \geq \delta$, taking expectations and using the symmetry $\mathbb{E}[\varepsilon_A] = \mathbb{E}[\varepsilon_B] = 1 - \overline{\mathsf{ps}}$ (which follows by exchanging the roles of the two attributes (p, q) of the cell in the average over all squares, which exchanges the two pairs of sides) yields (12). □

Remark 2 (Role of non-degeneracy). The factor $1/\sin^2\vartheta$ reflects the purely geometric obstruction: if ℓ_1 and ℓ_3 are nearly collinear ($\vartheta \approx 0$ or π), the four points $f_\theta(x^{(u,v)})$ for $(u, v) \in \{(a, c), (a, d), (b, c), (b, d)\}$ lie nearly on a line, and one can have $\mathsf{ps}_A = \mathsf{ps}_B = 1$ (all sides parallel) without $\mathsf{sq} = 0$ (the quadrilateral is a degenerate "squashed" shape, not a parallelogram). The condition $\sin\vartheta \geq \delta > 0$ rules out this degeneracy: it requires that *changing the first attribute* and *changing the second* displace the representation in genuinely different directions, which is a natural diversity condition on f_θ.

Combining the two above results we get the following:

Theorem 8 (Bounds between P-score and square defect). We have

$$2\,\delta_{\text{norm}}^2\,(1 - \overline{\mathsf{ps}}\,) \;\leq\; \overline{\mathsf{sq}^2} \;\leq\; \frac{48C^2}{\delta_{\text{angle}}^2}\,(1 - \overline{\mathsf{ps}}\,),$$

in which δ_{norm} is a lower bound on the sidelengths of compositional tuples C is an upper bound on $\|f_\theta(x)\|$, and δ_{angle} is a lower bound on the sine of the angle between sides of compositional tuples.

Hence under both nondegeneracy conditions $\overline{\mathsf{sq}^2} \to 0$ if and only if $\overline{\mathsf{ps}} \to 1$.

Remark 3. Note that random encoders in high dimension will tend to have bounds $\delta_{\text{norm}} > 0$ due to generic injectivity, and for $\delta_{\text{angle}} > 0$ based on the fact that, for independent random embeddings in large dimension, two sides of a cell sharing a vertex have cosine concentrating around $1/2$, i.e. an angle of about $60°$, so that $\sin\vartheta \approx 0.87$.

C.5 Transfer Across the Orthotopic Split: Proofs

This subsection proves Theorem 2 and Corollary 2. We use the notation of Section 3.2: $\varphi(y) := \mathbb{E}[f_\theta(x) \mid y]$; each value set is partitioned as $Y_m = E_m \sqcup H_m$ into a nonempty set of easy values and a possibly empty set of hard values (if $H_m = \emptyset$, the stratum $S_1^{(m)}$ is empty and is omitted from every maximum over strata below); $T(y) := \{m : y_m \in H_m\}$; the training set $S = S_0 \sqcup S_1^{(1)} \sqcup \cdots \sqcup S_1^{(M)}$ with strata $S_0 = \{y : |T(y)| = 0\}$ and $S_1^{(m)} = \{y : T(y) = \{m\}\}$, and the test set $U = \{y : |T(y)| \geq 2\}$; the additive class $\mathcal{A}$ and the defects $\varepsilon_{\text{train}}, \varepsilon_{\text{test}}$ of (4). We write $\mu_m := \text{Unif}(E_m)$ and $\nu_m := \text{Unif}(H_m)$, and for a set of coordinates $A \subseteq \{1, \ldots, M\}$ we let $\mu_A := \bigotimes_{m \in A} \mu_m$. No assumption is made on the sizes $|E_m|, |H_m|$ beyond $E_m \neq \emptyset$; ν_m is only used for $m \in T(y)$, where $H_m \neq \emptyset$.

Splicing and averaging. For $y \in Y$, $A \subseteq \{1, \dots, M\}$ and $v \in \prod_{m \in A} Y_m$, let $y[A \to v]$ denote y with the coordinates in A overwritten by v. For $g : Y \to \mathbb{R}^n$, the easy-averaging operator is

$$(\mathrm{Av}_A\, g)(y) \;:=\; \frac{1}{\prod_{m \in A} |E_m|} \sum_{v \,\in\, \prod_{m \in A} E_m} g\big(y[A \to v]\big), \tag{14}$$

a deterministic function of y that overwrites the coordinates in A with easy values in all possible ways and averages; $\mathrm{Av}_\emptyset = \mathrm{Id}$, and $\mathrm{Av}_A\, g$ does not depend on y_A. In terms of these operators, the imagined representation (5) reads

$$\Lambda\varphi(y) \;=\; \sum_{m \in T(y)} \big(\mathrm{Av}_{T(y)\setminus\{m\}}\, \varphi\big)(y) \;-\; \big(|T(y)| - 1\big)\, \big(\mathrm{Av}_{T(y)}\, \varphi\big)(y), \tag{15}$$

because the anchor $o = y[T(y) \to W]$ with $W \sim \mu_{T(y)}$ gives $\mathbb{E}_o\, \varphi(o) = (\mathrm{Av}_{T(y)}\varphi)(y)$, and $\varphi(o[m \to y_m])$ does not depend on the easy value drawn for coordinate m, so that $\mathbb{E}_o\, \varphi(o[m \to y_m]) = (\mathrm{Av}_{T(y)\setminus\{m\}}\varphi)(y)$. We extend Λ to all of Y by setting $\Lambda\varphi := \varphi$ on S, and write $R := \mathrm{Id} - \Lambda$, so that term (I) of Theorem 2, the imagination gap, is $\mathbb{E}_U \|R\varphi\|$. Regrouping (15) recovers the reading of Λ given in Section 3.2:

$$\Lambda\varphi(y) \;=\; \underbrace{\big(\mathrm{Av}_{T(y)}\varphi\big)(y)}_{\text{all-easy baseline}} \;+\; \sum_{m \in T(y)} \underbrace{\Big[\big(\mathrm{Av}_{T(y)\setminus\{m\}}\varphi\big)(y) - \big(\mathrm{Av}_{T(y)}\varphi\big)(y)\Big]}_{\text{increment of attribute } m \text{ alone}}, \tag{16}$$

i.e., it starts from the all-easy baseline and adds the separate one-at-a-time increments of the hard values of y, which is precisely the information displayed by the $c = 1$ training set.

Lemma 9 (Exactness, support, mass). (i) Λ is linear, and $\Lambda h = h$ on U for every $h \in \mathcal{A}$.

(ii) Every configuration at which φ is evaluated in (15) lies in S: the m-th term of the first group evaluates φ only on $S_1^{(m)}$, the second group only on S_0. Hence $\Lambda\varphi$ depends on the restriction $\varphi|_S$ alone.

(iii) The coefficients of (15) have total absolute mass $2|T(y)| - 1$ and sum to 1.

Proof. (ii) and (iii) are read off (15): in the m-th term of the first group only the coordinate m keeps a hard value, while in the second group no coordinate does; the two groups carry absolute masses $|T(y)|$ and $|T(y)| - 1$, and the signed sum of the coefficients is $|T(y)| - (|T(y)| - 1) = 1$.

(i) Linearity is clear. Let $h = \sum_m \alpha_m \in \mathcal{A}$ and write $\bar\alpha_m := \mathbb{E}_{v \sim \mu_m}\, \alpha_m(v)$. By (14), for any $A \subseteq T(y)$,

$$\big(\mathrm{Av}_A h\big)(y) \;=\; \sum_{m \in A} \bar\alpha_m \;+\; \sum_{m \notin A} \alpha_m(y_m).$$

Abbreviating $T := T(y)$ and $j := |T(y)|$, this gives

$$\begin{aligned}\Lambda h(y) = \sum_{m \in T} \Big[\alpha_m(y_m) + \sum_{m' \in T\setminus\{m\}} \bar\alpha_{m'} + \sum_{m' \notin T} \alpha_{m'}(y_{m'})\Big] \\ - (j-1)\Big[\sum_{m' \in T} \bar\alpha_{m'} + \sum_{m' \notin T} \alpha_{m'}(y_{m'})\Big].\end{aligned}$$

Each $\bar\alpha_{m'}$ with $m' \in T$ appears with coefficient $j - 1$ in the first group and $-(j-1)$ in the second, and each $\alpha_{m'}(y_{m'})$ with $m' \notin T$ appears with coefficient $j - (j-1) = 1$; what remains is $\sum_{m \in T} \alpha_m(y_m) + \sum_{m' \notin T} \alpha_{m'}(y_{m'}) = h(y)$. □

The averaging in (14) — as opposed to reconstructing from a single fixed easy baseline — makes the evaluation points of Λ *exactly uniform* on the training strata. This is the content of the next lemma; it needs no assumption on the proportions of easy and hard values, because the strata $S_1^{(m)}$ are treated separately.

Lemma 10 (Pushforward). Let $Y^\star \sim \mathrm{Unif}(U)$; given $Y^\star$, let $m^\star$ be uniform on $T(Y^\star)$ and, independently, $V^\star \sim \mu_{T(Y^\star)\setminus\{m^\star\}}$ and $W^\star \sim \mu_{T(Y^\star)}$. Then, conditionally on $(T(Y^\star), m^\star)$,

$$Z_1 := Y^\star\big[T(Y^\star) \setminus \{m^\star\} \to V^\star\big] \;\sim\; \mathrm{Unif}\big(S_1^{(m^\star)}\big),$$

and, conditionally on $T(Y^\star)$,

$$Z_0 := Y^\star\big[T(Y^\star) \to W^\star\big] \;\sim\; \mathrm{Unif}(S_0).$$

Proof. The test set is the disjoint union $U = \bigsqcup_{|T|\geq 2} U_T$ of the boxes $U_T := \prod_{m\in T} H_m \times \prod_{m\notin T} E_m$, so conditionally on $T(Y^\star) = T$ the configuration $Y^\star$ is uniform on U_T, i.e., its coordinates are independent with law ν_m for $m \in T$ and μ_m for $m \notin T$. Overwriting the coordinates in $T \setminus \{m^\star\}$ by independent uniform easy values leaves coordinate $m^\star$ with law $\nu_{m^\star}$ and every other coordinate with law $\mu_{m'}$, independently; this is the uniform law on $S_1^{(m^\star)} = H_{m^\star} \times \prod_{m'\neq m^\star} E_{m'}$. Overwriting all of T leaves independent $\mu_{m'}$ coordinates, the uniform law on $S_0 = \prod_{m'} E_{m'}$. □

Proof of Theorem 2. The support claim and the identity $\Lambda h = h$ on U are Lemma 9(ii) and (i). For the bound, fix $h \in \mathcal{A}$. On U, since $\Lambda h = h$ and Λ is linear,

$$\varphi - h \;=\; (\varphi - \Lambda\varphi) + \Lambda(\varphi - h).$$

Writing $g := \varphi - h$, the triangle inequality applied to (15) gives, pointwise on U,

$$\|\Lambda g(y)\| \;\leq\; \sum_{m\in T(y)} \big(\mathrm{Av}_{T(y)\setminus\{m\}}\|g\|\big)(y) \;+\; \big(|T(y)|-1\big)\big(\mathrm{Av}_{T(y)}\|g\|\big)(y).$$

Writing $\sum_{m\in T(y)} = |T(y)|\,\mathbb{E}_{m^\star}$ and each Av as an expectation over $V^\star$, $W^\star$ as in Lemma 10, and taking the expectation over $Y^\star \sim \mathrm{Unif}(U)$,

$$\mathbb{E}_U\|\Lambda g\| \;\leq\; \mathbb{E}\Big[\,|T(Y^\star)|\,\mathbb{E}\big[\|g(Z_1)\| \,\big|\, T(Y^\star), m^\star\big]\Big] \;+\; \mathbb{E}\Big[\big(|T(Y^\star)|-1\big)\,\mathbb{E}\big[\|g(Z_0)\| \,\big|\, T(Y^\star)\big]\Big].$$

By Lemma 10 the inner conditional expectations equal $\mathbb{E}_{S_1^{(m^\star)}}\|g\| \leq \max_m \mathbb{E}_{S_1^{(m)}}\|g\|$ and $\mathbb{E}_{S_0}\|g\|$ respectively, both bounded by $\max_\sigma \mathbb{E}_\sigma\|g\|$, so

$$\mathbb{E}_U\|\Lambda g\| \;\leq\; \big(\bar\jmath + (\bar\jmath - 1)\big)\max_\sigma \mathbb{E}_\sigma\|\varphi - h\| \;=\; (2\bar\jmath - 1)\max_\sigma \mathbb{E}_\sigma\|\varphi - h\|.$$

Hence, for every $h \in \mathcal{A}$,

$$\mathbb{E}_U\|\varphi - h\| \;\leq\; \mathbb{E}_U\|\varphi - \Lambda\varphi\| + (2\bar\jmath - 1)\max_\sigma \mathbb{E}_\sigma\|\varphi - h\|, \tag{17}$$

a per-comparator form of the theorem that is used in Appendix C.6. Conversely, on U we have $\varphi - \Lambda\varphi = (\varphi - h) - \Lambda(\varphi - h)$, and the same bound on $\mathbb{E}_U\|\Lambda(\varphi - h)\|$ gives $\mathbb{E}_U\|\varphi - \Lambda\varphi\| \leq \mathbb{E}_U\|\varphi - h\| + (2\bar\jmath - 1)\max_\sigma \mathbb{E}_\sigma\|\varphi - h\|$, which together with (17) is the two-sided statement; for a minimizer $h^\star$ of $\varepsilon_{\mathrm{train}}$ (which exists: the restriction map $h \mapsto h|_S$ is injective on $\mathcal{A}$, since an additive function vanishing on S vanishes on Y, every attribute value occurring in S; hence $h \mapsto \max_\sigma \mathbb{E}_\sigma\|h\|$ is a norm on the finite-dimensional space $\mathcal{A}$, the objective $h \mapsto \max_\sigma \mathbb{E}_\sigma\|\varphi - h\|$ is continuous and coercive, and it attains its infimum) the last maximum is $\varepsilon_{\mathrm{train}}(\varphi)$. Taking h near-optimal for $\varepsilon_{\mathrm{train}}$ in (4) and bounding $\varepsilon_{\mathrm{test}}(\varphi) \leq \mathbb{E}_U\|\varphi - h\|$ yields (6). Finally $\bar\jmath \leq M$ since $|T(y)| \leq M$. □

Remark 4 (The constant $\bar\jmath$). $\bar\jmath = \mathbb{E}_U[|T(y)|]$ is determined by the split alone. In the special case of a *balanced* split, $|H_m|/|Y_m| = p$ for all m, the number of hard coordinates J of a uniform configuration of Y is $\mathrm{Binomial}(M, p)$ and $\bar\jmath = \mathbb{E}[J \mid J \geq 2] = Mp\big(1 - (1-p)^{M-1}\big)/\big(1 - (1-p)^M - Mp(1-p)^{M-1}\big)$, which tends to 2 as $p \to 0$ and is close to Mp only when Mp is large. In our benchmarks the proportions differ across attributes (e.g., one hard value out of 3, 3, 2 and 8 on CLEVR), and $\bar\jmath$ is simply computed by counting. The constant $2\bar\jmath - 1$ is the average ℓ^1 mass of the coefficients of Λ, and the bound on $\mathbb{E}_U\|\Lambda g\|$ is attained (take $g = v$ on the strata $S_1^{(m)}$ and $g = -v$ on S_0); whether (6) itself is tight is not established.

Remark 5 (Relation to classical decompositions and prior extrapolation results). In the language of functional ANOVA, Λ is the first-order truncation of the anchored ANOVA (cut-HDMR) decomposition of φ in the hard coordinates, with the uniform distribution on easy values as anchor measure (Rabitz & Aliş, 1999; Kuo et al., 2010); Proposition 14 below states it as a projection onto additive functions and as a best additive estimate from training data.

Extrapolation to unseen combinations from additive structure is also the mechanism behind additive decoders (Lachapelle et al., 2023), whose additivity is architectural, behind the extrapolation guarantee

of CRM to the affine hull of the training groups (Mahajan et al., 2025) (the $c = 1$ test set lies inside it), and behind the analyses of Dong & Ma (2023) and Wiedemer et al. (2023). What the present analysis adds is the exact pushforward to the training strata with constant $2\bar{j} - 1$ and no assumption on the split, the two-sided identification of the test representation distance of the additive extension with held-out cell defects (Lemma 13), and the rigidity notion ι^H of Appendix C.7.

It remains to bound $\varepsilon_{\text{train}}$ by the imagination residuals on training cells, which is the content of the following corollary, stated in Section 3.2 in words.

Corollary 2 (*The worst-case imagination residual controls the training defect*). If every cell contained in S has imagination residual at most ϵ, then $\varepsilon_{\text{train}}(\varphi) \le \binom{M}{2}\epsilon$, and hence

$$\varepsilon_{\text{test}}(\varphi) \;\le\; \mathbb{E}_U\big\|\varphi - \Lambda\varphi\big\| \;+\; (2\bar{j} - 1)\binom{M}{2}\epsilon. \tag{18}$$

The first ingredient is the observation that the $c = 1$ training set is star-shaped, so that Corollary 1 applies to it.

Lemma 11 (Stratified approximate separability on the $c = 1$ training set). The training set S is star-shaped about every $o \in S_0$. Consequently, if every compositional cell whose four configurations lie in S satisfies $\|\mathsf{sq}\,\varphi\| \le \epsilon^\star$, then there exists a *single* $h \in \mathcal{A}$ with

$$\big\|\varphi(y) - h(y)\big\| \;\le\; \binom{M}{2}\epsilon^\star \qquad \text{for every } y \in S;$$

in particular, $\varepsilon_{\text{train}}(\varphi) \le \binom{M}{2}\epsilon^\star$.

Proof. Let $o \in S_0$ and $y \in S$. A configuration $y' \in \mathrm{Box}_o(y)$ has $y'_m \in \{y_m, o_m\}$ with o_m easy, so its hard coordinates form a subset of $T(y)$; hence $|T(y')| \le |T(y)| \le 1$ and $y' \in S$. Thus S is star-shaped about o, and Corollary 1 yields the single comparator $h = \sum_m \alpha_m$ with the stated pointwise bound on S. Since the bound is pointwise on $S = S_0 \sqcup \bigsqcup_m S_1^{(m)}$, every stratum expectation in (4) is bounded by it, giving $\varepsilon_{\text{train}}(\varphi) \le \binom{M}{2}\epsilon^\star$. □

The second ingredient converts the imagination residual of an arbitrary module into a square-defect bound.

Proof of Corollary 2. The hypothesis of the corollary implies $\|\mathsf{sq}\,\varphi\| \le \epsilon$ on every cell contained in S. Lemma 11 with $\epsilon^\star := \epsilon$ gives $\varepsilon_{\text{train}}(\varphi) \le \binom{M}{2}\epsilon$, and substituting into (6) yields (18). □

The corollary requires *every* training cell to have residual at most ϵ, whereas the imagination loss is an average: by Cauchy–Schwarz, $\mathbb{E}_{\text{cell}}\|\mathsf{sq}\,\varphi\| \le \sqrt{\mathcal{L}_{\text{img}}}$ over the sampled cells, which bounds the mean defect but not the maximum. The following averaged version replaces the maximum by the mean, at the price of a constant that depends only on the split and on the sampler.

Proposition 12 (Averaged control of the training defect). For $o \in S_0$ and $y \in S$, the proof of Corollary 1 bounds $\|\varphi(y) - h_o(y)\|$, where $h_o \in \mathcal{A}$ is the comparator built from the reference point o, by the sum of the defects of $\binom{M}{2}$ explicit cells $C_1(o, y), \ldots, C_{\binom{M}{2}}(o, y)$ contained in $\mathrm{Box}_o(y) \subseteq S$. Cells are taken *unoriented*, i.e. identified by the attribute pair, the two unordered value pairs and the context, since the square defect of a cell does not depend on its orientation up to sign. Let μ be the law on unoriented training cells induced by the sampler of Algorithm 2, and let

$$\kappa \;:=\; \max_{\sigma}\; \max_{k}\; \sup_{C}\; \frac{\mathbb{P}_{o\sim\mathrm{Unif}(S_0),\, y\sim\mathrm{Unif}(\sigma)}\big[C_k(o, y) = C\big]}{\mu(C)},$$

the supremum running over non-degenerate cells (degenerate ones have zero defect). Then

$$\varepsilon_{\text{train}}(\varphi) \;\le\; \binom{M}{2}\kappa\, \mathbb{E}_{C\sim\mu}\|\mathsf{sq}\,\varphi(C)\| \;\le\; \binom{M}{2}\kappa\,\sqrt{\mathcal{L}_{\text{img}}}.$$

Proof. Let $\bar{h} := \mathbb{E}_{o\sim\mathrm{Unif}(S_0)} h_o$, which lies in $\mathcal{A}$ since $\mathcal{A}$ is a linear space. For each stratum σ, by Jensen's inequality and the pointwise bound, $\mathbb{E}_\sigma\|\varphi - \bar{h}\| \le \mathbb{E}_o\mathbb{E}_\sigma\|\varphi - h_o\| \le \sum_k \mathbb{E}_{o,y}\|\mathsf{sq}\,\varphi(C_k(o,y))\| \le \binom{M}{2}\,\kappa\,\mathbb{E}_{C\sim\mu}\|\mathsf{sq}\,\varphi(C)\|$, by the definition of κ. Taking the maximum over σ bounds $\varepsilon_{\mathrm{train}}(\varphi)$, with Cauchy–Schwarz gives the second inequality. □

The constant κ is computable by counting from the split and the sampler; it can be large when the training strata are much larger than the set of cells the sampler favors, in which case the worst-case form of Corollary 2 is the informative one.

The imagination gap as a cell defect. The imagination gap of Theorem 2 is itself a square defect, evaluated on cells that straddle the split.

Lemma 13 (The imagination gap is an average of held-out cell defects). Let $y \in U$ with $T(y) = \{p, q\}$. Then

$$R\varphi(y) = \varphi(y) - \Lambda\varphi(y) = \mathbb{E}_{e_p\sim\mu_p,\ e_q\sim\mu_q}\Big[\mathsf{sq}\,\varphi\big(y[p,q\to e_p,e_q],\ y[p\to e_p],\ y[q\to e_q],\ y\big)\Big],$$

i.e., the average, over the easy reference values (e_p, e_q), of the defect of the cell with values $\{e_p, y_p\}$ for attribute p and $\{e_q, y_q\}$ for attribute q, in the context of y. In each such cell the three corners other than y have at most one hard value and lie in S, while the target y lies in U. Consequently

$$\|R\varphi(y)\| \le \max_{e_p,e_q}\big\|\mathsf{sq}\,\varphi\big(y[p,q\to e_p,e_q],\ y[p\to e_p],\ y[q\to e_q],\ y\big)\big\|.$$

Proof. For $T(y) = \{p, q\}$, (15) reads $\Lambda\varphi(y) = (\mathrm{Av}_{\{q\}}\varphi)(y) + (\mathrm{Av}_{\{p\}}\varphi)(y) - (\mathrm{Av}_{\{p,q\}}\varphi)(y)$. Expanding the averages with (14), and inserting the dummy averages $\mathbb{E}_{e_p}$ in the first term and $\mathbb{E}_{e_q}$ in the second,

$$\varphi(y) - \Lambda\varphi(y) = \mathbb{E}_{e_p,e_q}\Big[\varphi(y) - \varphi(y[q\to e_q]) - \varphi(y[p\to e_p]) + \varphi(y[p,q\to e_p,e_q])\Big],$$

which is the stated average of square defects with the sign convention of (7); the corners $y[p, q \to e_p, e_q] \in S_0$, $y[p \to e_p] \in S_1^{(q)}$ and $y[q \to e_q] \in S_1^{(p)}$ lie in S. The inequality is Jensen's. □

The imagined representation as a best additive estimate. For every $|T(y)|$, $\Lambda\varphi(y)$ is a projection of φ onto additive functions, evaluated at y. Fix $y \in U$, write $T := T(y)$, and consider the local box $B_y := \prod_{m\in T}(E_m \cup \{y_m\})$, on which $\varphi_y(\omega) := \varphi(y[T\to\omega])$ varies the hard coordinates of y while the others stay fixed. Let $P := \bigotimes_{m\in T}\mathrm{Unif}(E_m)$, the law of the anchors. For $g : B_y \to \mathbb{R}^n$, $m \in T$ and $v \in E_m \cup \{y_m\}$, put $\mathbb{E}_P[g \mid \omega_m = v] := \mathbb{E}_{\omega\sim P}\, g(\omega[m\to v])$, which averages the *other* coordinates over easy values and is therefore defined also for the hard value $v = y_m$, and define

$$(\Pi_1 g)(\omega) := \mathbb{E}_P\, g + \sum_{m\in T}\Big(\mathbb{E}_P[g \mid \omega_m] - \mathbb{E}_P\, g\Big), \qquad \omega \in B_y.$$

Proposition 14 (The imagined representation as a projection).

(i) $\Lambda\varphi(y) = (\Pi_1\varphi_y)(y_T)$ and $R\varphi(y) = \big((\mathrm{Id} - \Pi_1)\varphi_y\big)(y_T)$.

(ii) Π_1 is a projection onto the additive functions on B_y: $\Pi_1 g$ is additive for every g, and $\Pi_1 h = h$ for every additive h. Its kernel consists of the pure interactions, the g with $\mathbb{E}_P g = 0$ and $\mathbb{E}_P[g \mid \omega_m = v] = 0$ for all $m \in T$ and v. On the all-easy part $\prod_{m\in T} E_m$ it is the $L^2(P)$-orthogonal projection onto additive functions.

(iii) (*Best additive estimate from training data.*) Let $h^{(0)} = c + \sum_{m\in T}\alpha_m$ be a least-squares additive fit of φ_y on the all-easy configurations $\prod_{m\in T} E_m$, which lie in S_0, and, for each $m \in T$, let a_m be the least-squares fit of the remaining effect of y_m on the configurations whose only hard coordinate is m, which lie in $S_1^{(m)}$:

$$a_m := \arg\min_a\ \mathbb{E}_{\omega\sim P}\Big\|\varphi_y(\omega[m\to y_m]) - c - \sum_{m'\in T\setminus\{m\}}\alpha_{m'}(\omega_{m'}) - a\Big\|^2.$$

Then $\Lambda\varphi(y) = c + \sum_{m\in T} a_m$, the prediction at y of the additive model fitted in these two stages.

(iv) (*Limit of orthogonal projections.*) For $\epsilon \in (0,1)$ let $\nu_\epsilon := \bigotimes_{m\in T}\big((1-\epsilon)\,\mathrm{Unif}(E_m) + \epsilon\,\delta_{y_m}\big)$ and let h_ϵ be the $L^2(\nu_\epsilon)$-orthogonal projection of φ_y onto additive functions on B_y. Then $h_\epsilon(y_T) \to \Lambda\varphi(y)$ as $\epsilon \to 0$.

Proof. (i) With $\omega \sim P$, the configuration $y[T \to \omega]$ is the anchor o of (5), so $\mathbb{E}_P\varphi_y = (\mathrm{Av}_T\varphi)(y)$ and $\mathbb{E}_P[\varphi_y \mid \omega_m = y_m] = (\mathrm{Av}_{T\setminus\{m\}}\varphi)(y)$. Hence $(\Pi_1\varphi_y)(y_T) = \sum_{m\in T}(\mathrm{Av}_{T\setminus\{m\}}\varphi)(y) - (|T|-1)(\mathrm{Av}_T\varphi)(y)$, which is (15); the second identity follows since $R = \mathrm{Id} - \Lambda$. (ii) $\Pi_1 g$ is a constant plus a sum of functions of one coordinate each, hence additive. For $h = c + \sum_m a_m(\omega_m)$, with $\bar a_m := \mathbb{E}_{\mathrm{Unif}(E_m)}a_m$, we have $\mathbb{E}_P h = c + \sum_m \bar a_m$ and $\mathbb{E}_P[h \mid \omega_m = v] = c + a_m(v) + \sum_{m'\neq m}\bar a_{m'}$, so $\Pi_1 h = h$; thus $\Pi_1^2 = \Pi_1$, and the kernel is as stated. On $\prod_m E_m$, where P is a product probability measure with full support, Π_1 is the first-order Hoeffding (ANOVA) truncation, which is the $L^2(P)$-orthogonal projection onto additive functions. (iii) A least-squares fit with an intercept reproduces the mean, so $c + \sum_{m'}\bar\alpha_{m'} = \mathbb{E}_P\varphi_y$, and the normal equation of a_m gives $a_m = \mathbb{E}_P[\varphi_y \mid \omega_m = y_m] - c - \sum_{m'\neq m}\bar\alpha_{m'}$. Summing, $c + \sum_m a_m = \sum_m \mathbb{E}_P[\varphi_y \mid \omega_m = y_m] - (|T|-1)\,\mathbb{E}_P\varphi_y$, which is $\Lambda\varphi(y)$ by (i); the result does not depend on the choice among least-squares solutions of the first stage. (iv) Since ν_ϵ is a product probability measure with full support on B_y, h_ϵ is the first-order Hoeffding truncation with respect to ν_ϵ, $h_\epsilon(y_T) = \mathbb{E}_{\nu_\epsilon}\varphi_y + \sum_m\big(\mathbb{E}_{\nu_\epsilon}[\varphi_y \mid \omega_m = y_m] - \mathbb{E}_{\nu_\epsilon}\varphi_y\big)$. Both expectations are polynomial in ϵ and converge, as $\epsilon \to 0$, to $\mathbb{E}_P\varphi_y$ and $\mathbb{E}_P[\varphi_y \mid \omega_m = y_m]$, so $h_\epsilon(y_T) \to (\Pi_1\varphi_y)(y_T) = \Lambda\varphi(y)$ by (i). □

The imagined representation is therefore the best additive estimate of $\varphi(y)$ available from the training split: exactly, in the two-stage least-squares sense of (iii), and as the limit of L^2-best additive estimates under a reference measure on which hard values are rare, which is the situation the $c = 1$ split creates. The imagination gap is the error of this estimate at unseen compositions; $R\varphi(y)$ collects the interactions of order two and higher among the hard coordinates (for $|T(y)| = 2$, Lemma 13 writes it as held-out cell defects), and it is in every case computable from φ on $S \cup \{y\}$, hence measurable post hoc on held-out configurations. The map $\varphi \mapsto \Lambda\varphi$ is local, since it conditions on the easy coordinates of y; it coincides with the additive model fitted on the whole training split only for additive encoders, and Theorem 2 controls the difference in general.

Measuring the imagination gap. Algorithm 1 computes the normalized imagination gap $\rho_{(\mathrm{I})}$ reported in Table 3. Line 19 is (5) with the expectation over anchors replaced by an average over enumerated or sampled anchors, and $\mathbf{b}$ estimates $\mathbb{E}_o\varphi(o) = (\mathrm{Av}_{T(y)}\varphi)(y)$ with the same anchors. The denominator is the distance from the average anchor, i.e. the displacement produced by the hard values of y, which is what the imagined representation has to reproduce; the ratio is invariant to rescaling and translation of the representation, and vanishes exactly when the imagination gap does. When anchors are sampled rather than enumerated, both numerator and denominator are biased upward as K decreases, so the direction of the bias of the ratio is not known, and anchors should be enumerated whenever feasible.

C.6 From Additive Geometry to Exact-Match Accuracy

The results so far control distances in representation space. This subsection connects them to the metric we report, exact-match accuracy under per-attribute affine readouts. Two facts are established: additivity alone is not sufficient for any readout to succeed (Proposition 15), and additivity together with linearly independent factor directions is sufficient for affine readouts to be exact and robust (Theorem 17); the margin argument that converts a representation-level bound into an error bound is Lemma 16.

Proposition 15 (Additivity alone does not imply separability). Let $M = 2$, $Y_1 = Y_2 = \{1,2\}$, and $\varphi(v,w) := (v+w)\,e$ for a fixed unit vector $e \in \mathbb{R}^n$. Then $\varphi \in \mathcal{A}$, so $\mathsf{sq}\,\varphi$ vanishes on every cell, yet $\varphi(1,2) = \varphi(2,1)$: the two configurations with different labels share a representation, and no classifier, linear or otherwise, can predict both attributes correctly on all of Y.

Proof. $\varphi(v,w) = \alpha_1(v) + \alpha_2(w)$ with $\alpha_1(v) = ve$, $\alpha_2(w) = we$, so $\varphi \in \mathcal{A}$ and its square defect vanishes by the linearity remark of Appendix C.1. The collision is immediate. □

Algorithm 1 Normalized imagination gap $\rho_{(\mathrm{I})}$ (term (I) of Theorem 2)

Require: In-support representations and labels $\{(z_i, y_i)\}$ (training split, and ID-validation if drawn from the same support); held-out test representations and labels $\{(z'_i, y'_i)\}$, from the same checkpoint and layer as the P-score

Require: Easy and hard values E_m, H_m of each attribute m; number of Monte Carlo anchors K; enumeration budget $N_{\max}$

Ensure: $\rho_{(\mathrm{I})} = \mathbb{E}_U\|\varphi(y) - \Lambda\varphi(y)\| \,/\, \mathbb{E}_U\|\varphi(y) - \mathbb{E}_o\varphi(o)\|$

$\varphi_{\mathrm{in}}(y) \leftarrow$ mean of z_i over in-support images with $y_i = y$
$\varphi_{\mathrm{test}}(y) \leftarrow$ mean of z'_i over test images with $y'_i = y$
$\mathcal{N} \leftarrow \emptyset$, $\mathcal{D} \leftarrow \emptyset$
for each test configuration y **do**
  $T \leftarrow \{m : y_m \in H_m\}$
  if $|T| < 2$ **then continue** ▷ $y \notin U$
  end if
  if $\prod_{m\in T} |E_m| \le N_{\max}$ **then**
    $\mathcal{E} \leftarrow \prod_{m\in T} E_m$ ▷ exact enumeration of anchors
  else
    $\mathcal{E} \leftarrow K$ tuples e with $e_m \sim \mathrm{Unif}(E_m)$ independently for $m \in T$
  end if
  $\mathbf{a} \leftarrow 0, \quad \mathbf{b} \leftarrow 0, \quad n \leftarrow 0$
  for each $e \in \mathcal{E}$ **do**
    $o \leftarrow y[T \to e]$ ▷ anchor, $o \in S_0$
    $s_m \leftarrow o[m \to y_m]$ for each $m \in T$ ▷ one hard value, $s_m \in S_1^{(m)}$
    if o or some s_m is absent from φ_{in} **then continue**
    end if
    $\mathbf{a} \leftarrow \mathbf{a} + \varphi_{\mathrm{in}}(o) + \sum_{m\in T}\big(\varphi_{\mathrm{in}}(s_m) - \varphi_{\mathrm{in}}(o)\big)$
    $\mathbf{b} \leftarrow \mathbf{b} + \varphi_{\mathrm{in}}(o), \quad n \leftarrow n + 1$
  end for
  if $n = 0$ **then continue**
  end if
  $\mathbf{a} \leftarrow \mathbf{a}/n$ ▷ imagined representation $\Lambda\varphi(y)$, eq. (5)
  $\mathbf{b} \leftarrow \mathbf{b}/n$ ▷ average anchor $\mathbb{E}_o\varphi(o)$
  add $\|\varphi_{\mathrm{test}}(y) - \mathbf{a}\|$ to $\mathcal{N}$ and $\|\varphi_{\mathrm{test}}(y) - \mathbf{b}\|$ to $\mathcal{D}$; record $|T|$ and the coverage $n/|\mathcal{E}|$
end for
return $\rho_{(\mathrm{I})} \leftarrow \mathrm{mean}(\mathcal{N})/\,\mathrm{mean}(\mathcal{D})$ ▷ ratio of means; optionally per value of $|T|$

In the example the two factor directions coincide, $\alpha_1(Y_1)$ and $\alpha_2(Y_2)$ both lying on the line $\mathbb{R}e$. What a linear readout needs, in addition to additivity, is that the factor directions be independent; this is the geometric condition identified by Uselis et al. (2026) as necessary for compositional generalization of linear readouts on vision embeddings, and Theorem 17 shows it is also sufficient once the representation is additive.

Lemma 16 (Margin). Let $r_m(z) = \arg\max_{v\in Y_m} s_{m,v}(z)$ with affine scores $s_{m,v}(z) = \langle w_{m,v}, z\rangle + b_{m,v}$, and let $L_m := \max_{v\neq v'} \|w_{m,v} - w_{m,v'}\|$. If $s_{m,u}(z) - \max_{v\neq u} s_{m,v}(z) \ge \gamma$ and $\|z' - z\|\, L_m < \gamma$, then $r_m(z') = u$.

Proof. For $v \neq u$, $s_{m,u}(z') - s_{m,v}(z') = s_{m,u}(z) - s_{m,v}(z) + \langle w_{m,u} - w_{m,v}, z' - z\rangle \ \ge\ \gamma - L_m\|z' - z\| > 0$. □

Corollary 3 (Exact-match transfer). Let attribute m be predicted by $r_m(z) = \arg\max_{v\in Y_m} s_{m,v}(z)$ with affine scores $s_{m,v}(z) = \langle w_{m,v}, z\rangle + b_{m,v}$, and put $L := \max_m \max_{v\neq v'} \|w_{m,v} - w_{m,v'}\|$. Suppose there is an additive comparator $h \in \mathcal{A}$ on which the readout has margin $\gamma > 0$ over the test split: $s_{m,y_m}(h(y)) - \max_{v\neq y_m} s_{m,v}(h(y)) \ge \gamma$ for every $y \in U$ and every m. Then the exact-match error on U satisfies

$$\mathbb{P}_U\big[\, r(\varphi(y)) \neq y \,\big] \ \le\ \frac{L}{\gamma}\Big(\mathbb{E}_U\|\varphi - \Lambda\varphi\| + (2\bar{\jmath} - 1)\max_{\sigma}\mathbb{E}_\sigma\|\varphi - h\|\Big).$$

In particular, if the margin condition holds for a minimizer $h^\star$ of (4) (one exists, since $\mathcal{A}$ is a finite-dimensional linear space and $h \mapsto \max_\sigma \mathbb{E}_\sigma\|\varphi - h\|$ is continuous and coercive, see the proof of Theorem 2), then

$$\mathbb{P}_U\big[\, r(\varphi(y)) \neq y \,\big] \;\le\; \frac{L}{\gamma}\Big(\underbrace{\mathbb{E}_U\|\varphi - \Lambda\varphi\|}_{\text{(I)}} + \underbrace{(2\bar{\jmath} - 1)\,\varepsilon_{\mathrm{train}}(\varphi)}_{\text{(II)}}\Big),$$

the form quoted in Section 3.2.

Proof of Corollary 3. Let $y \in U$. By Lemma 16 applied with $z = h(y)$, $z' = \varphi(y)$ and $u = y_m$, the readout predicts every attribute of y correctly whenever $\|\varphi(y) - h(y)\| < \gamma/L$. Hence the exact-match error event is contained in $\{\|\varphi(y) - h(y)\| \ge \gamma/L\}$, whose probability under $\mathrm{Unif}(U)$ is at most $(L/\gamma)\,\mathbb{E}_U\|\varphi - h\|$ by Markov's inequality. Theorem 2, in the form (17), bounds $\mathbb{E}_U\|\varphi - h\|$ by $\mathbb{E}_U\|\varphi - \Lambda\varphi\| + (2\bar{\jmath} - 1)\max_\sigma \mathbb{E}_\sigma\|\varphi - h\|$. For $h = h^\star$ the last maximum equals $\varepsilon_{\mathrm{train}}(\varphi)$ by definition. □

Theorem 17 (Additive and orthogonal representations admit exact, robust affine readouts). Let $h = \sum_m \alpha_m \in \mathcal{A}$. Write $\bar{\alpha}_m := |Y_m|^{-1}\sum_{v \in Y_m} \alpha_m(v)$, $u_m(v) := \alpha_m(v) - \bar{\alpha}_m$, $V_m := \mathrm{span}\{u_m(v) : v \in Y_m\}$, $c := \sum_m \bar{\alpha}_m$, and

$$\delta_m := \min_{v \neq v'} \|\alpha_m(v) - \alpha_m(v')\|, \qquad D_m := \max_{v, v'} \|\alpha_m(v) - \alpha_m(v')\|.$$

Assume within-factor separation, $\delta_m > 0$ for all m, and cross-factor orthogonality, $V_m \perp V_{m'}$ for $m \neq m'$. Let P_m be the orthogonal projector onto V_m and define the affine scores

$$s_{m,v}(z) := \big\langle u_m(v),\, P_m(z - c)\big\rangle - \tfrac{1}{2}\|u_m(v)\|^2, \qquad r_m(z) := \arg\max_{v \in Y_m} s_{m,v}(z).$$

Then:

(i) (*Exactness on the full grid.*) $r_m(h(y)) = y_m$ for every $y \in Y$ and every m.

(ii) (*Margin.*) $s_{m,y_m}(h(y)) - \max_{v \neq y_m} s_{m,v}(h(y)) \ge \delta_m^2/2$ for every $y \in Y$, and $L_m = D_m$.

(iii) (*Robustness.*) If $\|z - h(y)\| < \delta_m/2$ then $r_m(z) = y_m$. In particular, for any encoder φ and any $Q \subseteq Y$,

$$\mathbb{P}_{y \sim \mathrm{Unif}(Q)}\big[r(\varphi(y)) \neq y\big] \;\le\; \frac{2}{\min_m \delta_m}\,\mathbb{E}_Q\|\varphi - h\|.$$

Proof. Since $u_m(v) \in V_m$ and P_m is self-adjoint, $s_{m,v}$ is affine in z with weight vector $w_{m,v} = P_m u_m(v) = u_m(v)$, so $L_m = \max_{v \neq v'}\|u_m(v) - u_m(v')\| = D_m$. For $z = h(y)$, $h(y) - c = \sum_{m'} u_{m'}(y_{m'})$ and orthogonality gives $P_m(h(y) - c) = u_m(y_m)$. Hence

$$s_{m,v}(h(y)) = \langle u_m(v), u_m(y_m)\rangle - \tfrac{1}{2}\|u_m(v)\|^2 = \tfrac{1}{2}\|u_m(y_m)\|^2 - \tfrac{1}{2}\|u_m(v) - u_m(y_m)\|^2,$$

which is uniquely maximized at $v = y_m$ with margin $\frac{1}{2}\min_{v \neq y_m}\|u_m(v) - u_m(y_m)\|^2 \ge \delta_m^2/2$, since $\|u_m(v) - u_m(y_m)\| = \|\alpha_m(v) - \alpha_m(y_m)\|$. This proves (i) and (ii). For (iii), with $\eta := z - h(y)$ and $v \neq y_m$,

$$\begin{aligned} s_{m,y_m}(z) - s_{m,v}(z) &= \tfrac{1}{2}\|u_m(v) - u_m(y_m)\|^2 + \langle u_m(y_m) - u_m(v), P_m\eta\rangle \\ &\ge \|u_m(v) - u_m(y_m)\|\Big(\tfrac{1}{2}\|u_m(v) - u_m(y_m)\| - \|P_m\eta\|\Big), \end{aligned}$$

which is positive when $\|P_m\eta\| \le \|\eta\| < \delta_m/2$. The error bound follows from Markov's inequality applied to the event $\{\|\varphi(y) - h(y)\| \ge \min_m \delta_m/2\}$. □

Corollary 4 (Exact-match transfer with explicit constants). Under the assumptions of Theorem 17, with its readout,

$$\mathbb{P}_U\big[r(\varphi(y)) \neq y\big] \;\le\; \frac{2}{\min_m \delta_m}\Big(\mathbb{E}_U\|\varphi - \Lambda\varphi\| + (2\bar{\jmath} - 1)\max_\sigma \mathbb{E}_\sigma\|\varphi - h\|\Big).$$

Proof. Combine Theorem 17(iii) with $Q = U$ and (17). □

Remark 6 (Orthogonality versus independence; relation to the nondegeneracy constants). Orthogonality can be relaxed to linear independence of the subspaces V_m (i.e., $V_m \cap \sum_{m' \neq m} V_{m'} = \{0\}$): replacing P_m by the oblique projector Π_m onto V_m along $\bigoplus_{m' \neq m} V_{m'}$ leaves (i) and the margin in (ii) unchanged, changes the weight vectors to $\Pi_m^\top u_m(v)$ and hence L_m, and replaces $\delta_m/2$ by $\delta_m/(2\|\Pi_m\|)$ in (iii). The same hypotheses supply the nondegeneracy constants of Theorem 8: for an additive h the sides of every cell are factor displacements $\alpha_p(b) - \alpha_p(a)$ and $\alpha_q(d) - \alpha_q(c)$, so $\delta_{\mathrm{norm}} \geq \min_m \delta_m$, and under orthogonality adjacent sides are perpendicular, so $\delta_{\mathrm{angle}} = 1$.

Remark 7 (Relation to additive-energy classifiers). CRM (Mahajan et al., 2025) assumes that the energy of a configuration decomposes over attributes, $\mathcal{E}(x, y) = \sum_m \mathcal{E}_m(x, y_m)$, and derives its extrapolation guarantee and its bias correction B^* from that decomposition holding on unseen groups. Additivity of the encoder is the representation-level counterpart. With per-attribute affine readouts $s_{m,v}(z) = \langle w_{m,v}, z\rangle + b_{m,v}$ and an additive encoder $\varphi = \sum_{m'} \alpha_{m'}$, the joint score of a candidate label y' at the input $x(y)$ is

$$\sum_m s_{m,y'_m}(\varphi(y)) = \sum_m \sum_{m'} \big\langle w_{m,y'_m}, \alpha_{m'}(y_{m'})\big\rangle + \sum_m b_{m,y'_m},$$

a sum of terms each depending on a single label coordinate and, through the encoder, on the generative factors of the input only additively; under the cross-factor orthogonality of Theorem 17 and the readout constructed there the double sum collapses to $\sum_m \langle u_m(y'_m), u_m(y_m)\rangle$, a separable energy. This is the additive-energy form under which B^* is derived. Note, however, that a per-attribute readout yields energies additive in the labels whatever the encoder computes, and that the validity of B^* rests on CRM's assumption about the data distribution, which additivity of the encoder does not establish. The correspondence is therefore a heuristic: it suggests that the correction should become less harmful once the encoder is trained toward additivity, which is what Section 5 observes, but it does not prove it.

C.7 On Higher-Order Interactions and Architectural Rigidity

This subsection gives a more detailed account of term (I) in (6), and of the structural restrictions required to control it. Recall that $R := \mathrm{Id} - \Lambda$. The imagined representation $\Lambda\varphi(y)$ starts from an all-easy baseline and adds the separate one-at-a-time increments of the hard values in y. Thus $R\varphi(y) = \varphi(y) - \Lambda\varphi(y)$ is the component not determined by those first-order effects, including interactions of order two and higher among the hard attributes (for $|T(y)| = 2$, exactly the held-out cell defect of Lemma 13). Since $\Lambda\varphi$ depends on $\varphi|_S$ alone, these interactions are not displayed by the $c = 1$ training set.

First we formalize the claim from section 3.2 that capacity can not control term (I):

Proposition 18 (Capacity does not control term (I)). Suppose

$$|U| > 1 + \sum_{m=1}^{M} \big(|Y_m| - 1\big).$$

Then for every $B > 0$ there is a two-element encoder class $\mathcal{F} = \{\varphi^0, \varphi^1\}$—so with VC, pseudo- and fat-shattering dimension at most 1, covering numbers at most 2 at every scale, and empirical Rademacher complexity at most $\|\varphi^1\|_\infty \sqrt{2 \ln 2/N}$ on every sample of any size, whether drawn from S or from all of Y—such that

$$\varepsilon_{\mathrm{train}}(\varphi) = 0 \quad \text{for both } \varphi \in \mathcal{F}, \qquad \max_{\varphi \in \mathcal{F}} \varepsilon_{\mathrm{test}}(\varphi) \geq B.$$

Proof. The restriction map $\mathcal{A} \to (\mathbb{R}^n)^U$ has image whose scalar coordinate projections have dimension at most $1 + \sum_m (|Y_m| - 1)$, since an additive map is determined by the M maps α_m modulo the $(M-1)$-dimensional gauge of shifting constants between them. Under the hypothesis this image is a proper subspace in at least one output coordinate, so some $g : U \to \mathbb{R}^n$ has $\varepsilon_{\mathrm{test}}(g) > 0$. Put $\varphi^0 := 0$, and $\varphi^1 := \lambda g$ on U with $\varphi^1 := 0$ on S, where $\lambda := B/\varepsilon_{\mathrm{test}}(g)$. Both vanish on S, hence $\varepsilon_{\mathrm{train}} = 0$; and $\varepsilon_{\mathrm{test}}(\varphi^1) = B$. The complexity claims are immediate for a two-element class, the last by Massart's finite-class lemma. □

The hypothesis is mild: $|U|$ grows with the size of the configuration grid, whereas the right-hand side grows only linearly in M and in the number of attribute values.

The counterexample shows that small hypothesis-class cardinality and VC-type combinatorial capacity alone do not link behavior on S to behavior on U. The missing assumption must be structural: it must prevent functions that agree on the complete training support from varying independently on unseen combinations. We call such a restriction a *rigidity class*.

Definition 2 (Rigidity class)**.** A class $\mathcal{F}$ of encoders is (r, r_0)-*rigid* for a defect functional $\mathcal{E}$ depending on $\varphi|_S$ only, and a constant $r_0 \geq 0$ of the class, if

$$\mathbb{E}_U \big\| R\varphi \big\| \;\leq\; r\,\mathcal{E}(\varphi) + r_0 \qquad \text{for every } \varphi \in \mathcal{F}.$$

By Theorem 2,

$$\varepsilon_{\text{test}}(\varphi) \;\leq\; r\,\mathcal{E}(\varphi) + r_0 + (2\bar{\jmath} - 1)\varepsilon_{\text{train}}(\varphi)$$

for such a class.

Block-structured encoders. We consider encoders whose output is a concatenation of attribute-specific blocks, $\varphi = (\varphi_1, \ldots, \varphi_M)$ with $\varphi_m(y) \in \mathbb{R}^{n_m}$ and block m read by the head for attribute m; AIN and ED are of this form. (For a shared vector z with per-attribute readouts $z \mapsto W_m z$, the same results apply to the block encoder $(W_1\varphi, \ldots, W_M\varphi)$.) An architecture with an attribute-specific block is designed so that φ_m responds to y_m and ignores the remaining attributes. Perfect invariance is too strong to assume; two defects quantify the deviation from it, and it matters which one an assumption is placed on.

Definition 3 (Invariance defects)**.** For a block-structured encoder $\varphi = (\varphi_1, \ldots, \varphi_M)$ and $m \in \{1, \ldots, M\}$, put

$$\iota_m(\varphi) \;:=\; \sup\Big\{ \|\varphi_m(y) - \varphi_m(y')\| :\; y, y' \in Y,\; y_m = y'_m \Big\},$$
$$\iota_m^H(\varphi) \;:=\; \sup\Big\{ \big\|\varphi_m(y) - \big(\mathrm{Av}_{\{m'\}}\varphi_m\big)(y)\big\| :\; y \in Y,\; m' \neq m,\; y_{m'} \in H_{m'} \Big\},$$

and $\iota := (\sum_m \iota_m^2)^{1/2}$, $\iota^H := (\sum_m (\iota_m^H)^2)^{1/2}$. The *full* defect ι_m is the largest change in block m caused by varying attributes other than m. The *hard* defect ι_m^H is the largest deviation of block m, when some other attribute m' carries a *hard* value, from the average of block m over the easy values of m'; by Jensen's inequality $\iota_m^H \leq \iota_m$. In particular ι_m^H places no constraint on how block m depends on the easy values of other attributes.

The full defect makes term (I) irrelevant, because it bounds the test defect directly: this is the regime in which the architecture alone suffices.

Proposition 19 (Full block invariance suffices alone). For every block-structured φ, $\varepsilon_{\text{test}}(\varphi) \leq \iota(\varphi)$ and $\mathbb{E}_U \|R\varphi\| \leq 2\bar{\jmath}\,\iota(\varphi)$.

Proof. Fix a reference $o \in Y$ and let $\bar{\varphi}_m(y) := \varphi_m(y_m, o_{-m})$. Then $\bar{\varphi}_m$ depends only on y_m, so the concatenation $\bar{\varphi} := (\bar{\varphi}_1, \ldots, \bar{\varphi}_M)$ lies in $\mathcal{A}$, and $\|\varphi_m - \bar{\varphi}_m\| \leq \iota_m$ pointwise by Definition 3. Hence $\|\varphi - \bar{\varphi}\| \leq \iota$ pointwise on Y, which gives $\varepsilon_{\text{test}}(\varphi) \leq \iota$. For the residual, $R\bar{\varphi} = 0$ because $\Lambda h = h$ for every $h \in \mathcal{A}$, so $R\varphi = R(\varphi - \bar{\varphi})$; since Λ has coefficient mass $2|T(y)| - 1$ (Lemma 9), $\|R\varphi(y)\| \leq (1 + 2|T(y)| - 1)\,\iota = 2|T(y)|\iota$, and averaging over U gives the claim. □

Note that the second bound is never the operative one: inserting it into (6) gives $2\bar{\jmath}\,\iota + (2\bar{\jmath} - 1)\varepsilon_{\text{train}} \geq 4\iota$, which is weaker than $\varepsilon_{\text{test}} \leq \iota$ for every value of $\varepsilon_{\text{train}}$. Under full block invariance the imagination objective adds nothing to the guarantee. The hard defect is the right quantity for the complementary regime: it controls term (I) but, on its own, does not control $\varepsilon_{\text{test}}$.

Theorem 20 (Hard block invariance is a rigidity class). For every block-structured encoder φ,

$$\|R\varphi(y)\| \;\leq\; 2|T(y)|\,\iota^H(\varphi) \quad \text{for all } y \in U, \qquad \text{hence} \qquad \mathbb{E}_U \big\| R\varphi \big\| \;\leq\; 2\bar{\jmath}\,\iota^H(\varphi).$$

Consequently the class $\mathcal{F}_\tau^H$ of encoders with $\iota^H(\varphi) \leq \tau$ is $(0, 2\bar{\jmath}\tau)$-rigid, and Theorem 2 and Corollary 2 give, for $\varphi \in \mathcal{F}_\tau^H$,

$$\varepsilon_{\text{test}}(\varphi) \;\leq\; 2\bar{\jmath}\tau + (2\bar{\jmath} - 1)\,\varepsilon_{\text{train}}(\varphi) \;\leq\; 2\bar{\jmath}\tau + (2\bar{\jmath} - 1)\binom{M}{2}\epsilon. \tag{19}$$

Proof. Fix $y \in U$, write $T := T(y)$, $j := |T(y)|$, and for $A \subseteq T$ and each block m put $F_m(A) := (\mathrm{Av}_A \varphi_m)(y)$. By (15), $R\varphi_m(y) = F_m(\emptyset) - \sum_{m' \in T} F_m(T \setminus \{m'\}) + (j-1)F_m(T)$.

Step 1: one coordinate at a time. Let $A \subseteq B \subseteq T$ with $m \notin B \setminus A$. Then $\|F_m(A) - F_m(B)\| \le |B \setminus A|\, \iota_m^H$. Indeed, order $B \setminus A = \{m'_1, \dots, m'_k\}$ and set $A_i := A \cup \{m'_1, \dots, m'_i\}$. For each i,

$$F_m(A_{i-1}) - F_m(A_i) = \mathrm{Av}_{v \sim \mu_{A_{i-1}}} \Big[\varphi_m\big(y[A_{i-1} \to v]\big) - \big(\mathrm{Av}_{\{m'_i\}} \varphi_m\big)\big(y[A_{i-1} \to v]\big) \Big],$$

and in every configuration $y[A_{i-1} \to v]$ the coordinate $m'_i \in T \setminus A_{i-1}$ still carries its hard value $y_{m'_i}$ while $m'_i \neq m$; so each bracket has norm at most ι_m^H by Definition 3, and Jensen's inequality bounds the average by the same. Summing over i gives the claim.

Step 2: blocks whose attribute is easy in y ($m \notin T$).

$$R\varphi_m(y) = \big[F_m(\emptyset) - F_m(T)\big] - \sum_{m' \in T} \big[F_m(T \setminus \{m'\}) - F_m(T)\big],$$

and Step 1 applies to every bracket (no bracket removes coordinate m): the first has norm at most $j\, \iota_m^H$ and each of the j others at most ι_m^H. Hence $\|R\varphi_m(y)\| \le 2j\, \iota_m^H$.

Step 3: blocks whose attribute is hard in y ($m \in T$).

$$R\varphi_m(y) = \big[F_m(\emptyset) - F_m(T \setminus \{m\})\big] - \sum_{m' \in T \setminus \{m\}} \big[F_m(T \setminus \{m'\}) - F_m(T)\big],$$

where we used $-\sum_{m' \in T} F_m(T \setminus \{m'\}) = -F_m(T \setminus \{m\}) - \sum_{m' \neq m} F_m(T \setminus \{m'\})$ and $(j-1)F_m(T) = \sum_{m' \in T \setminus \{m\}} F_m(T)$. Step 1 applies to every bracket: the first removes the $j-1$ coordinates of $T \setminus \{m\}$, none equal to m, and each of the $j-1$ others removes the single coordinate $m' \neq m$. Hence $\|R\varphi_m(y)\| \le 2(j-1)\, \iota_m^H \le 2j\, \iota_m^H$.

Step 4: conclusion. Since $\|R\varphi(y)\|^2 = \sum_m \|R\varphi_m(y)\|^2 \le 4j^2 \sum_m (\iota_m^H)^2$, we get $\|R\varphi(y)\| \le 2|T(y)|\iota^H$; averaging over U gives $\mathbb{E}_U \|R\varphi\| \le 2\bar{\jmath}\, \iota^H$, and (19) follows from (6) and (18). □

Remark 8 (Neither term of (19) controls $\varepsilon_{\mathrm{test}}$ alone). The two terms of (19) bound genuinely different parts of the encoder. (a) $\varepsilon_{\mathrm{train}} = 0$ is compatible with arbitrarily large $\varepsilon_{\mathrm{test}}$ (Proposition 18), so the objective alone certifies nothing on U. (b) $\iota^H = 0$ is compatible with $\varepsilon_{\mathrm{test}} > 0$ as soon as $M \ge 3$: take $M = 3$, let $g : Y_1 \times E_3 \to \mathbb{R}^{n_1}$ be non-additive on $H_1 \times E_3$ (which requires $|H_1| \ge 2$ and $|E_3| \ge 2$), and set $\varphi_1(y) := g(y_1, y_3)$ if $y_3 \in E_3$ and $\varphi_1(y) := |E_3|^{-1} \sum_{e \in E_3} g(y_1, e)$ if $y_3 \in H_3$, with φ_2, φ_3 depending on y_2, y_3 only. Then $\iota^H = 0$, but on the configurations $(h_1, h_2, e_3) \in U$ the first block equals $g(h_1, e_3)$ and is not additive, so $\varepsilon_{\mathrm{test}}(\varphi) > 0$; correspondingly $\varepsilon_{\mathrm{train}}(\varphi) > 0$, because the same non-additive dependence is displayed on S_0 and $S_1^{(1)}$. The dependence of a block on the *easy* values of other attributes is visible on the training split and is what the imagination loss penalizes through $\varepsilon_{\mathrm{train}}$; the dependence on *hard* values of other attributes in the presence of a second hard value is never visible during training and is what the architecture must supply through ι^H. This is the formal content of the claim that a structural bias and a recombination objective are complementary rather than redundant. (For $M = 2$ every configuration of U has both coordinates hard, and $\iota^H = 0$ then forces additivity on U; the complementarity is a phenomenon of three or more attributes.)

AIN provides a direct example of the block-structured restriction above. Its attribute-specific components are intended to make the block associated with attribute m respond primarily to y_m, thereby reducing both invariance defects. What Theorem 20 isolates is that only the hard part ι^H has to come from the architecture: in the idealized limit $\iota^H = 0$ the imagination gap vanishes and (19) reduces to $\varepsilon_{\mathrm{test}} \le (2\bar{\jmath} - 1)\varepsilon_{\mathrm{train}}$, a purely training-split quantity controlled by the imagination loss.

CRM imposes a related restriction at the score or readout level. Its additive energy decomposition requires the score assigned to a new composition to be assembled from attribute-wise terms learned on observed configurations, rather than permitting an arbitrary correction that appears only when hard attributes are combined. It is therefore an example of the broader rigidity principle in Definition 2. The bound of Theorem 20, however, applies to a block-structured representation such as AIN's;

applying it to CRM's encoder would additionally require showing that CRM's representation satisfies the hard-invariance hypothesis.

This also explains why the imagination objective need not yield the same gain for every encoder family. The most the objective can remove from (19) is the term $(2\bar{\jmath}-1)\varepsilon_{\text{train}}$. If ι^H is large, the term $2\bar{\jmath}\,\iota^H$ dominates and driving the training defect to zero moves the bound little. If the encoder is already fully invariant (ι small), Proposition 19 bounds $\varepsilon_{\text{test}}$ without any objective. The objective therefore pays off in the intermediate regime: a structured encoder with small hard defect whose square defect on training cells is not spontaneously small. This is the regime in which an architectural restriction and LATTICE's recombination objective act as complementary ingredients.

Remark 9 (Mechanism: what is proven and what is measured). Summarizing the chain of Section 3 and this appendix. (i) On training cells, the fixed I_{ana} makes the imagination loss penalize square defects of the encoder. Vanishing defects make its mean representation additive on the training split, and small defects give an approximate version (Theorem 4, Corollary 1, Lemma 11). (ii) Additivity on S is transported to U by the analogy rule (Theorem 2) at the price of the imagination gap, term (I), which losses evaluated on the training split alone cannot guarantee to be small (Proposition 18). This gap is the part of the representation that appears only when two hard values meet; a shared encoder, which computes an attribute's contribution with the same filters wherever that attribute occurs, has no capacity dedicated to such conjunction-specific features, and architectures with attribute-specific blocks are designed to bias against them, which Theorem 20 quantifies through ι^H; we do not measure ι^H, and the imagination gap only bounds it from below. (iii) We do not prove that LATTICE reduces the imagination gap; we measure it. It is computable on held-out configurations (Lemma 13, Algorithm 1), Table 3 reports it in normalized form next to the P-score on held-out compositions, and Figure 2 shows individual cells with a held-out target. (iv) Given (I), (II) and a readout with a margin, Corollary 3 turns the representation-level bound into a bound on exact-match error. What remains an inductive-bias argument rather than a theorem is the step from (i) to a small term (I); this is the limitation of the analysis, and the reason the experiments report the held-out geometry directly.

C.8 Further Properties of the Imagination Objective

We collect three elementary facts about $\mathcal{L}_{\text{img}}$ that bear on questions raised by the experiments: what the objective sees when the representation carries nuisance structure, what it enforces when episodes are not compositional cells, and why it can be reduced without effect on backbones whose readout is invariant to the scale of the representation.

Proposition 21 (The objective sees only the non-additive component). Let $\varphi = h + \eta$ with $h \in \mathcal{A}$. Then $\mathsf{sq}\,\varphi = \mathsf{sq}\,\eta$ on every cell, so the imagination loss of the analogical module equals $\mathbb{E}_{\text{cell}}\|\mathsf{sq}\,\eta\|^2$, and all results of Appendices C.3 and C.5 apply with $\epsilon = \sup_{\text{cells}}\|\mathsf{sq}\,\eta\|$. In particular, any component of η that depends only on the context r of a cell (attributes not varied within the episode) cancels exactly, and a component that is uncorrelated with the varied factors enters only through its cell defect.

Proof. sq is linear and vanishes on $\mathcal{A}$ (Appendix C.1). If $\eta(y) = g(r)$ on a cell with context r, the four values coincide and $\mathsf{sq}\,\eta = g(r) - g(r) - g(r) + g(r) = 0$. □

Under compositional sampling the four members of an episode share their context, so nuisance variation tied to the context is invisible to the objective; nuisance variation that varies within a cell contributes to ϵ and degrades the guarantees continuously rather than breaking them. Nuisance that is *correlated* with the varied factors is not distinguished from the factors themselves by any of the results above.

Proposition 22 (Without compositional structure the objective is a variance penalty). (i) Let $Z \subseteq \mathbb{R}^n$ and suppose $\mathsf{sq}(z_1, z_2, z_3, z_4) = 0$ for every ordered quadruple $(z_1, z_2, z_3, z_4) \in Z^4$. Then $|Z| = 1$.

(ii) If z_1, z_2, z_3, z_4 are i.i.d. with covariance Σ, then $\mathbb{E}\|\mathsf{sq}(z_1, z_2, z_3, z_4)\|^2 = 4\,\mathrm{tr}\,\Sigma$.

Proof. (i) Taking $z_1 = z_2$ gives $z_4 = z_3$ for all $z_3, z_4 \in Z$. (ii) The coefficients $(1, -1, -1, 1)$ sum to zero, so the mean cancels and the four independent centered terms contribute $\mathrm{tr}\,\Sigma$ each. □

When the episode members are drawn i.i.d. from the training set, as in the ablation without compositional sampling, the imagination loss is therefore four times the total variance of the representation: a generic shrinkage regularizer whose unconstrained minimizer is a collapsed representation and which carries no information about which factor changed. Its interaction with $\mathcal{L}_{\mathrm{cls}}$ is that of any variance penalty, and its effect can be expected to depend on the dataset rather than on compositional structure.

Proposition 23 (Scale-invariant readouts admit a free descent direction). Suppose the classification loss is invariant under positive rescaling of the representation, $\mathcal{L}_{\mathrm{cls}}(t\, f_\theta) = \mathcal{L}_{\mathrm{cls}}(f_\theta)$ for all $t > 0$, as is the case for readouts based on cosine similarity to a fixed codebook, and let $f^{(t)} := t\, f_\theta$ (realizable by rescaling the last linear layer). Then

$$\mathcal{L}_{\mathrm{cls}}(f^{(t)}) = \mathcal{L}_{\mathrm{cls}}(f_\theta), \qquad \mathcal{L}_{\mathrm{img}}(f^{(t)}) = t^2\, \mathcal{L}_{\mathrm{img}}(f_\theta),$$

while every prediction and the P-score of every cell are unchanged. Hence $\mathcal{L}_{\mathrm{total}}(f^{(t)})$ decreases strictly as $t \downarrow 0$ and $\inf_t \mathcal{L}_{\mathrm{total}}(f^{(t)}) = \mathcal{L}_{\mathrm{cls}}(f_\theta)$: the imagination loss can be driven to zero along a direction that alters neither the geometry measured by the P-score nor the classifier.

Proof. sq is linear, so $\|\mathsf{sq}(tz)\|^2 = t^2\|\mathsf{sq}(z)\|^2$; the P-score is a cosine between displacements and is invariant under $z \mapsto tz$; the invariance of $\mathcal{L}_{\mathrm{cls}}$ is the hypothesis. □

With a linear readout, rescaling the representation changes the logits only if the readout is held fixed: a jointly trained readout can compensate, since $(t\, f_\theta, W/t)$ leaves the logits and $\mathcal{L}_{\mathrm{cls}}$ unchanged while scaling $\mathcal{L}_{\mathrm{img}}$ by t^2. The same direction is therefore available for the linear per-attribute readouts used in all our experiments (Section 4) unless the scale is controlled. We train without weight decay, so the objective alone does not fix the scale of the representation, and part of the reduction in $\mathcal{L}_{\mathrm{img}}$ may be a change of scale that is uninformative about compositional structure. This is why every geometric quantity we report, the P-score and $\rho_{(\mathrm{I})}$, is invariant to rescaling: a reduction of $\mathcal{L}_{\mathrm{img}}$ by shrinkage alone would not change them. The theoretical bounds are unaffected, since every inequality of Appendices C.2–C.5 is homogeneous of degree one in φ; only statements that a defect is "small" require a normalization.

Remark 10 (Block defect of per-attribute architectures). For a block-structured encoder, $\mathsf{sq}\,\varphi = (\mathsf{sq}\,\varphi_1, \ldots, \mathsf{sq}\,\varphi_M)$ on every cell. If block m depends on y_m only, then $\mathsf{sq}\,\varphi_m = 0$ on every cell: on cells that do not vary attribute m all four values coincide, and on cells varying (m, q) the two corners sharing each value of m cancel pairwise. In general $\mathsf{sq}\,\varphi_m$ is the defect of the block's dependence on the other attributes, so an architecture with one encoder per attribute has zero square defect if its blocks are invariant ($\iota = 0$ in Definition 3), not by construction; the converse fails, since a block depending additively on the other attributes also has zero defect. Having the highest validation P-score before any compositional objective is applied is the empirical signature of blocks that are close to invariant on the training support; a markedly lower test P-score suggests that this invariance does not fully extend to held-out combinations, i.e. that the hard defect ι^H of Definition 3 is not negligible; the imagination gap gives a lower bound on it, $\iota^H \geq \mathbb{E}_U\|R\varphi\|/(2\bar{\jmath})$ by Theorem 20.

D EXPERIMENTAL SETUP

D.1 ORTHOTOPIC EVALUATION FRAMEWORK

We adopt the Orthotopic evaluation framework (Camposampiero et al., 2025), a controlled benchmark for evaluating compositional generalization in fully supervised settings that allows for fair evaluation of compositional generalization methods. The framework constructs train–test splits by systematically holding out combinations of attribute values, thereby inducing different degrees of distribution shift while keeping the underlying prediction tasks fixed.

The protocol is governed by a difficulty parameter $c \in \{0, \dots, M\}$, where M denotes the number of tasks, or equivalently, the number of semantic factors considered in the dataset. For each task, attribute values are arbitrarily partitioned into *easy* and *hard* values. A sample is included in the training set if it contains at most c hard values across all tasks; otherwise, it is assigned to the test set. For example, consider a dataset with three tasks: *shape*, *color*, and *size*. Suppose their possible values are

$$\{\circ, \triangle, \square\}, \quad \{\textit{red}, \textit{blue}, \textit{green}\}, \quad \{\textit{small}, \textit{medium}, \textit{large}\},$$

and that $\{\square, \textit{green}, \textit{large}\}$ are selected as hard values. If $c = 1$, then a sample such as $\{\circ, \textit{red}, \textit{large}\}$ belongs to the training set, since it contains only one hard value. In contrast, $\{\circ, \textit{green}, \textit{large}\}$ is assigned to the test set, because it contains two hard values, exceeding the threshold $c = 1$.

Smaller values of c therefore induce stronger distribution shifts and more challenging generalization regimes. When $c = 0$, the model is evaluated on samples containing hard attribute values that were never observed during training. When $c = 1$, the model observes each hard value individually during training, but must generalize to novel combinations of multiple hard values at test time, which is the hardest setting for compositional generalization. As c increases, the split becomes progressively easier, since test examples share more attribute-value interactions with the training distribution. In this work, we focus exclusively on the $c = 1$ regime, following the disentangled compositional generalization setting.

D.2 DATASETS

We use the six benchmarks and the per-attribute easy/hard value assignments of Camposampiero et al. (2025); we refer to that work for the hard values used in each benchmark.

Cars3D (Reed et al., 2015). A synthetic dataset of 3D-rendered car images with controlled factors of variation, commonly used for disentanglement and viewpoint generalization. Typical factors include *car identity*, *azimuth*, and *elevation*.

dSprites (Matthey et al., 2017). A synthetic dataset of binary 64×64 images of simple 2D shapes, generated from the Cartesian product of latent factors such as *shape*, *scale*, *rotation*, *x-position*, and *y-position* (with *color* effectively constant in the standard version).

MPI3D (Gondal et al., 2019). A suite of datasets (*toy*, *realistic* and *real*) depicting physical solids attached to a robotic finger. Following Camposampiero et al. (2025), we use *MPI3D-real*, which consists of photographs of a real robot arm holding simple objects and is therefore the only genuinely real-world dataset among the six we evaluate. Its task-relevant generative factors are *object color*, *shape*, *camera height*, *background color*, and the arm's *x* and *y* positions.

CLEVR (Johnson et al., 2017; Camposampiero et al., 2025). Originally a synthetic visual question answering benchmark of rendered scenes with several objects. Following Camposampiero et al. (2025), we use a modified version in which each image contains a single object, designed for attribute classification: the task is to predict the object's attributes (e.g., *shape*, *color*, *material*).

I-RAVEN (Hu et al., 2021). (*Impartial-RAVEN*) An improved variant of the RAVEN dataset (Zhang et al., 2019) which itself is based on Raven's Progressive Matrices (Raven, 1938). This version mitigates known shortcut issues by constructing more balanced alternative answer sets, aiming for a fairer evaluation of *abstract visual reasoning*. Following Camposampiero et al. (2025),

we treat each panel of a matrix as an independent image, and the task is to predict the attributes of its object, *shape*, *size* and *color*; *angle* is excluded because of its ambiguity.

Shapes3D (Burgess & Kim, 2018). A synthetic dataset of 3D objects in a room (images of size 64×64) generated from independent factors such as *floor color*, *wall color*, *object color*, *shape*, *scale*, and *orientation*, with systematic combinations across these factors.

Training sets and coverage of compositional sampling. Table 4 reports the number of training images of each benchmark under the $c = 1$ split, and Figure 3 shows example images. On all six benchmarks, every training image belongs to at least one 2×2 cell whose four configurations lie in the training split, so compositional sampling (Algorithm 2) covers the whole training set.

Table 4: Training images per benchmark under the $c = 1$ split, and fraction of them that belong to at least one 2×2 cell with all four configurations in the training split.

	Cars3D	**dSprites**	**MPI3D**	**CLEVR**	**I-RAVEN**	**Shapes3D**
Training images	9,822	214,668	508,203	54,416	89,516	246,065
Coverage	100%	100%	100%	100%	100%	100%

D.3 Algorithms

Algorithm 2 Compositional Sampling for LATTICE

Require: Configuration set $Y = Y_1 \times \cdots \times Y_M$ and training split $S \subseteq Y$

1: **for** each compositional draw **do**
2: Choose two factor indices $p \neq q$
3: Sample $a \neq b$ from Y_p, $c \neq d$ from Y_q and a context r for all factors in $\{1, \ldots, M\} \setminus \{p, q\}$, such that all four configurations below lie in S
4: Define the compositional cell

$$y^{(a,c)} = (a, c, r), \quad y^{(a,d)} = (a, d, r), \quad y^{(b,c)} = (b, c, r), \quad y^{(b,d)} = (b, d, r)$$

5: Retrieve observations $x^{(a,c)}, x^{(a,d)}, x^{(b,c)}, x^{(b,d)}$ matching the corresponding attribute vectors
6: Return source tuple $\left(x^{(a,c)}, x^{(a,d)}, x^{(b,c)}\right)$ and target $x^{(b,d)}$
7: **end for**

D.4 Hyperparameters and Implementation

All experiments were run with a batch size of 1024 for all datasets, using the AdamW optimizer with a learning rate of 0.0001, without weight decay, and FP16 precision. All results are averaged over 5 seeds, rather than the 3 reported by Camposampiero et al. (2025), to reduce the uncertainty of the mean results, although the variability across seeds remains high. Following Camposampiero et al. (2025), for each run we select the checkpoint with the highest ID validation accuracy, taking the latest one in case of ties. The amount of training epochs is listed in Table 5. LATTICE introduces a single hyperparameter, the weight λ of the imagination loss in $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{img}}$. We swept $\lambda \in \{0, 0.5, 0.75, 1, 1.5, 2\}$ on AIN, where $\lambda = 0$ recovers training without the imagination loss; Table 2 (right) reports the averages and Table 10 the per-dataset breakdown. ID validation accuracy is saturated (close to 100%) for every λ and cannot discriminate between values, so the sweep is reported on test accuracy. Accuracy is flat across $[0.75, 2]$, and any λ in this range improves AIN by at least 7.8% over the baseline (58.77), so we did not tune λ per dataset or per backbone: all reported results use $\lambda = 1.5$ on all three architectures and all six benchmarks.

Implementation. Our implementation extends that of Camposampiero et al. (2025). When LATTICE is combined with CRM, the cross-entropy in $\mathcal{L}_{\text{cls}}$ is replaced by CRM's additive-energy cross-entropy, as for any auxiliary objective, so that $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CRM}} + \lambda \mathcal{L}_{\text{img}}$. CRM_{loss} differs from CRM only in the bias correction: the evaluation group set is the full Cartesian product and the test

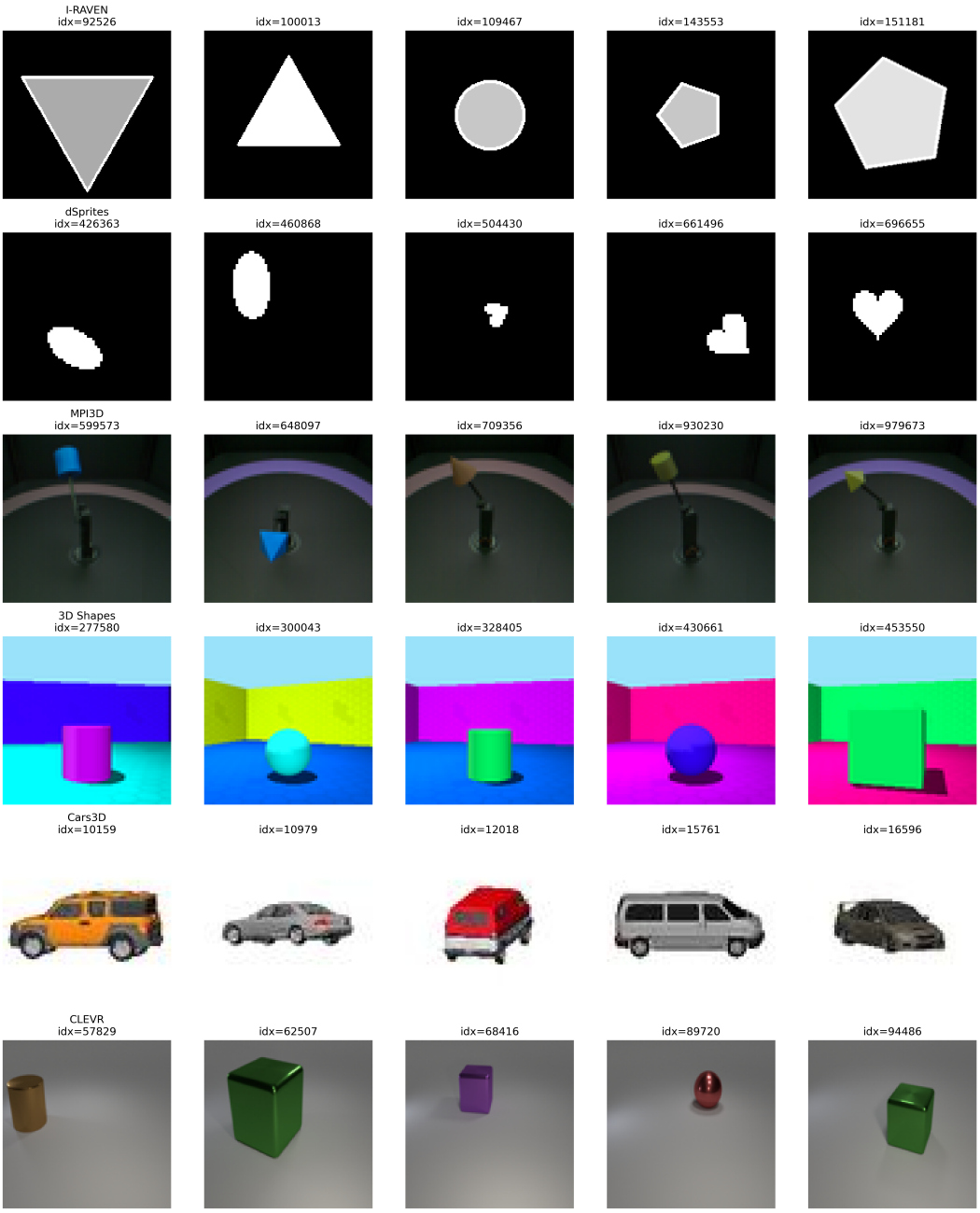

Figure 3: **Example images from each dataset.** I-RAVEN, dSprites, MPI3D, Shapes3D, Cars3D and CLEVR.

prior is uniform, so the joint CRM decision factorizes into the per-attribute argmax whenever B^* is constant, and any gap between CRM and CRM_{loss} is due to B^* alone.

Table 5: Hyperparameters used for reported results for different datasets.

Model	**Parameter**	**Cars3D**	**dSprites**	**MPI3D**	**CLEVR**	**I-RAVEN**	**Shapes3D**
All	No. Epochs	500	50	50	50	50	250
All + LATTICE	λ	1.5	1.5	1.5	1.5	1.5	1.5

Algorithm 3 LATTICE Training Step

Require: Mini-batch of compositional episodes $\mathcal{B} = \left\{ \left(x_i^{(u,v)}, y_i^{(u,v)} \right)_{(u,v)\in\{a,b\}\times\{c,d\}} \right\}_{i=1}^{B}$
Require: Encoder f_θ, classifier g_ω, imagination rule $\mathcal{I}$ (parameter-free)
Require: Loss weight $\lambda > 0$
$\mathcal{L}_{\text{cls}} \leftarrow 0$, $\mathcal{L}_{\text{img}} \leftarrow 0$
for $i = 1$ to B **do**
  for $(u, v) \in \{(a, c), (a, d), (b, c), (b, d)\}$ **do**
    $z_i^{(u,v)} \leftarrow f_\theta\left(x_i^{(u,v)}\right)$
    $\hat{y}_i^{(u,v)} \leftarrow g_\omega\left(z_i^{(u,v)}\right)$
    $\mathcal{L}_{\text{cls}} \leftarrow \mathcal{L}_{\text{cls}} + \ell_{\text{attr}}\left(\hat{y}_i^{(u,v)}, y_i^{(u,v)}\right)$
  end for
  $\tilde{z}_i^{(b,d)} \leftarrow \mathcal{I}\left(z_i^{(a,c)}, z_i^{(a,d)}, z_i^{(b,c)}\right)$
  $\mathcal{L}_{\text{img}} \leftarrow \mathcal{L}_{\text{img}} + \left\| \tilde{z}_i^{(b,d)} - z_i^{(b,d)} \right\|_2^2$
end for
$\mathcal{L}_{\text{cls}} \leftarrow \frac{1}{4B}\mathcal{L}_{\text{cls}}, \quad \mathcal{L}_{\text{img}} \leftarrow \frac{1}{B}\mathcal{L}_{\text{img}}$
$\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{cls}} + \lambda\,\mathcal{L}_{\text{img}}$
Update (θ, ω) with one gradient step on $\nabla\mathcal{L}_{\text{total}}$

D.5 Parameter Counts

Because the imagination loss is a parameter-free linear analogy, LATTICE leaves the parameter count unchanged, at training time as well as at inference. AIN does not expose an intermediate representation on its own, so all our AIN experiments — baseline and LATTICE alike — use a final linear readout, matching the encoder–classifier split that ResNet-18 and ED provide natively. Counts are therefore identical with and without LATTICE, and identical at training and inference, so Table 6 reports a single figure per model and dataset.

Table 6: Parameter count relative to ResNet-18 on the same dataset (%). All architectures use a linear per-attribute readout. LATTICE adds no parameters, so training and inference counts are identical and the LATTICE rows match their base exactly. CRM adds a per-group bias table $\hat{B}$, whose size scales with the number of observed groups: negligible on most benchmarks but 0.28M parameters (+2.5%) on MPI3D. ED replicates the full backbone once per attribute, so its cost grows with the number of attributes, from 3.1× on Cars3D and I-RAVEN (3 attributes) to 6.2× on MPI3D (6 attributes); AIN stays within 1.1× of ResNet-18 throughout.

Model	Cars3D	dSprites	MPI3D	CLEVR	I-RAVEN	Shapes3D
ResNet-18	100.00	100.00	100.00	100.00	100.00	100.00
ResNet-18 + CRM	100.09	100.05	102.49	100.00	100.00	100.16
ResNet-18 + LATTICE	100.00	100.00	100.00	100.00	100.00	100.00
ResNet-18 + CRM + LATTICE	100.09	100.05	102.49	100.00	100.00	100.16
AIN	105.60	103.88	107.09	105.33	103.38	106.14
AIN + CRM	105.69	103.93	109.58	105.33	103.38	106.30
AIN + LATTICE	105.60	103.88	107.09	105.33	103.38	106.14
AIN + CRM + LATTICE	105.69	103.93	109.58	105.33	103.38	106.30
ED	309.51	410.87	617.77	412.72	308.42	515.59
ED + CRM	309.59	410.92	620.26	412.72	308.42	515.75
ED + LATTICE	309.51	410.87	617.77	412.72	308.42	515.59
ED + CRM + LATTICE	309.59	410.92	620.26	412.72	308.42	515.75

D.6 Computational Resources

All experiments were run as single-GPU jobs with 4–8 CPU cores and 50 GB of RAM each. Jobs were scheduled on whichever machine was available, across three GPU types: NVIDIA H100 80GB, NVIDIA A40 48GB, and NVIDIA GeForce RTX 2080 SUPER; a single model family can therefore span several GPU types. Table 8 reports the median wall-clock time per run, measured from the training logs.

These times are not directly comparable across rows. Besides hardware, they depend on whether representation metrics such as the P-score were computed after every epoch, which can dominate the runtime on datasets trained for many epochs (Cars3D: 500 epochs; Shapes3D: 250). The batch size is the same throughout: every model sees 1024 images per step, which for LATTICE are drawn as 256 analogy quadruples, so the number of optimization steps is unchanged as well. LATTICE itself adds no parameters: the analogy loss is computed in closed form on representations already produced by the forward pass. The CRM_{loss} results reuse the CRM runs, since both share the trained model and differ only in the decision rule, and required no additional training.

Table 7 reports the total compute actually used, summing the measured wall-clock time of every run. It covers the main comparison (Tables 1 and 3), the λ sweep (Table 10), the sampling ablation (Table 2), and a set of re-runs of existing configurations with per-epoch representation metrics enabled, used only for the training-dynamics figures. It is not hardware-normalized.

Table 7: Total compute used, in raw GPU-hours summed over all runs; not hardware-normalized.

Model group	**Runs**	**H100**	**A40**	**RTX 2080S**
Baselines	90	259.2	–	122.9
+ CRM	90	283.9	178.9	–
+ LATTICE	90	223.2	63.1	–
+ CRM + LATTICE	90	313.0	103.2	–
Main comparison	360	1079.3	345.2	122.9
λ sweep ($\lambda \in \{0, 0.5, 0.75, 1, 2\}$)	150	242.5	361.5	–
Sampling ablation	120	125.8	248.6	–
Re-runs with per-epoch metrics	100	239.4	187.4	–
Total	730	1686.9	1142.7	122.9

Table 8: Median wall-clock training time per run, over 5 seeds. Superscripts give the GPU: ^{H} H100, ^{A} A40, ^{R} RTX 2080 SUPER. $^{\circ}$Runs trained without per-epoch representation metrics, which lowers their runtime; all other cells computed them after every epoch.

Model	**Cars3D**	**dSprites**	**MPI3D**	**CLEVR**	**I-RAVEN**	**Shapes3D**
ResNet-18	1h32m^{R}	7h24m^{R}	6h28m^{H}	5h50m^{R}	2h02m^{R}	7h50m^{R}
ResNet-18 + CRM	7h52m^{A}	1h27m^{A}	4h14m^{H}	3h08m^{H}	1h17m^{A}	1h41m^{A}
ResNet-18 + LATTICE	5h21m^{H}	1h00m^{H}	2h23m^{H}	3h48m^{H}	40m^{H}	1h01m^{H}
ResNet-18 + CRM + LATTICE	36m$^{\text{A}\circ}$	1h39m$^{\text{A}\circ}$	4h42m^{H}	3h34m^{H}	1h10m$^{\text{A}\circ}$	1h38m$^{\text{A}\circ}$
AIN	6h28m^{H}	1h42m^{H}	4h07m^{H}	6h55m^{H}	46m^{H}	1h58m^{H}
AIN + CRM	9h26m^{A}	6h03m^{A}	9h16m^{H}	7h15m^{H}	1h57m^{A}	5h58m^{A}
AIN + LATTICE	1h17m$^{\text{A}\circ}$	5h02m$^{\text{A}\circ}$	3h06m$^{\text{H}\circ}$	4h25m$^{\text{H}\circ}$	1h34m$^{\text{A}\circ}$	3h38m$^{\text{A}\circ}$
AIN + CRM + LATTICE	2h32m$^{\text{A}\circ}$	5h41m$^{\text{A}\circ}$	9h24m^{H}	7h22m^{H}	2h02m$^{\text{A}\circ}$	4h20m$^{\text{A}\circ}$
ED	6h18m^{H}	1h43m^{H}	5h35m^{H}	6h48m^{H}	46m^{H}	1h59m^{H}
ED + CRM	6h00m^{H}	1h52m^{H}	8h59m^{H}	12h03m^{H}	50m^{H}	3h53m^{H}
ED + LATTICE	6h36m^{H}	1h43m^{H}	3h51m^{H}	7h16m^{H}	44m^{H}	1h52m^{H}
ED + CRM + LATTICE	6h37m^{H}	2h10m^{H}	10h38m^{H}	15h07m^{H}	56m^{H}	4h55m^{H}

E Additional Accuracy Results

E.1 Per-Dataset Ablations

Table 2 in the main text reports the component and λ ablations averaged over benchmarks. Tables 9 and 10 give the corresponding per-dataset breakdowns.

Table 9: Ablation of AIN + LATTICE across all datasets. We remove the two components of LATTICE: compositional sampling, which draws the parallelogram quadruples, and the imagination loss, which imposes the latent parallelogram constraint. The imagination loss is the essential component: removing it costs accuracy on all six benchmarks, by up to 18.5% on I-RAVEN, and every dataset's best configuration retains it. The two components are strongly synergistic: relative to plain AIN, imagination alone contributes $+4.03$ and sampling alone $+0.85$, yet together they yield $+9.86$. Results are averaged over 5 seeds with standard error of the mean in parentheses.

Model	Cars3D	dSprites	MPI3D	CLEVR	I-RAVEN	Shapes3D	Avg
AIN + LATTICE	**56.72 (0.51)**	**67.25 (1.58)**	**57.92 (2.49)**	**60.50 (0.62)**	**76.49 (4.18)**	92.89 (1.22)	**68.63**
− sampling	47.92 (0.39)	61.60 (1.27)	54.65 (0.68)	50.78 (1.83)	67.75 (3.12)	**94.11 (1.68)**	62.80
− imagination	47.82 (0.39)	60.43 (2.99)	55.36 (1.16)	46.87 (3.75)	57.99 (2.34)	89.28 (1.27)	59.62
− both (AIN)	46.82 (0.89)	55.76 (1.19)	59.55 (2.91)	47.24 (2.41)	59.20 (1.92)	84.07 (2.46)	58.77

Table 10: Effect of the LATTICE loss weight λ on AIN. Test accuracy (%) with model selection based on ID performance, averaged over 5 seeds, with standard error of the mean in parentheses. $\lambda = 0$ disables the analogy loss while keeping the architecture, batch size, analogy-structured sampling and schedule of the $\lambda > 0$ runs, so the whole column is a controlled ablation of the objective alone. It therefore differs from the AIN row of Table 1 (58.77 on average), which is trained with i.i.d. sampling. Best per column in bold. Accuracy rises sharply up to $\lambda = 0.75$ and then plateaus, so the method is not sensitive to the exact weight.

λ	Cars3D	dSprites	MPI3D	CLEVR	I-RAVEN	Shapes3D	Avg
0	47.82 (0.39)	60.43 (2.99)	55.36 (1.16)	46.87 (3.75)	57.99 (2.34)	89.28 (1.27)	59.62
0.5	53.18 (0.93)	67.68 (1.27)	56.56 (0.80)	**61.20 (2.42)**	70.02 (4.36)	90.54 (1.97)	66.53
0.75	55.56 (0.91)	**68.77 (0.94)**	**61.97 (3.43)**	58.39 (2.84)	73.78 (2.43)	92.46 (1.00)	68.49
1	54.33 (1.22)	67.36 (1.21)	57.46 (2.68)	59.76 (1.83)	69.71 (3.61)	90.79 (1.45)	66.57
1.5	**56.72 (0.51)**	67.25 (1.58)	57.92 (2.49)	60.50 (0.62)	**76.49 (4.18)**	**92.89 (1.22)**	**68.63**
2	55.72 (0.68)	63.85 (2.26)	59.66 (2.85)	**61.20 (2.62)**	71.34 (2.47)	92.68 (0.99)	67.41

E.2 Parameterized Imagination Rule

Proposition 24 (*Unconstrained imagination imposes no structure on the encoder*). If φ is injective and $\varphi^{-1}(z)$ denotes the preimage of z, then the module $I^*(z_1, z_2, z_3) := \varphi\big(\psi(\varphi^{-1}(z_1), \varphi^{-1}(z_2), \varphi^{-1}(z_3))\big)$, with $\psi(y^{(a,c)}, y^{(a,d)}, y^{(b,c)}) := y^{(b,d)}$, satisfies $I^*\big(\varphi(y^{(a,c)}), \varphi(y^{(a,d)}), \varphi(y^{(b,c)})\big) = \varphi(y^{(b,d)})$ on every compositional cell, whether or not φ is additive; when each configuration is realized by a single image, its imagination loss is zero.

The module must therefore be too restricted to memorize the map from three embeddings to the fourth; the parameter-free analogy I_{ana} is the extreme case, in which the only way to reduce the loss is to move the encoder. Additivity of the encoder is loosely analogous to the additive-energy assumption of CRM (Mahajan et al., 2025); we use this only as a heuristic (Remark 7), since the validity of CRM's post-hoc bias correction B^* depends on its assumption about the data, not on the encoder.

Proposition 24 theoretically shows that a rule expressive enough to be learned can drive the imagination loss to zero without making the encoder additive. This section reports an experiment that we believe offers empirical support for that concern.

LATTICE implements the analogy in closed form: the rule has no parameters, and the loss directly penalizes the parallelogram residual $\|z^{(a,c)} - z^{(a,d)} - z^{(b,c)} + z^{(b,d)}\|^2$. Before settling on this form, we instantiated the rule as a learned network — a two-layer, four-head transformer encoder that receives the per-attribute representation pieces and is trained to predict the held-out corner of each analogy quadruple — and varied its width, using per-attribute piece dimensions of 64, 128 and 256.

Table 11 reports the results. The learned rule is worse than the parameter-free one on every backbone: -2.9% for ResNet-18 at its best width, -4.6 for AIN and -1.3 for ED. On ED it is also worse than not using LATTICE at all, and the gap widens as the network gets wider, from -2.5% at $d = 64$ to -4.5 at $d = 256$. On ED, capacity appears to hurt monotonically, whereas on AIN it has little effect and on ResNet-18 it helps (38.74 to 42.06) without reaching the parameter-free rule. We read the ED trend as consistent with Proposition 24: a rule with enough capacity may satisfy the analogy objective by compensating for representations that are not actually parallel, leaving the encoder's geometry unconstrained, whereas the closed-form residual can only be reduced by making the representations themselves additive. In the notation of Corollary 2, a learned rule has absorption $\Delta_{\mathcal{I}} > 0$, which loosens the certificate that the imagination residual provides on the training defect. We do not measure that mechanism directly, so we take these results as suggestive rather than conclusive. They do indicate, however, that the parameter-free formulation is not merely a simplification of a more expressive method.

Table 11: Test accuracy (%) of a learned imagination rule against the parameter-free analogy used by LATTICE, with model selection based on ID performance, averaged over 5 seeds, with standard error of the mean in parentheses. d is the per-attribute piece dimension of the learned rule. All variants are trained with compositional sampling. Base and LATTICE rows repeat Table 1. Best per backbone in bold. Shaded rows use LATTICE (ours).

Model	Cars3D	dSprites	MPI3D	CLEVR	I-RAVEN	Shapes3D	Avg
ResNet-18	33.12 (0.29)	21.30 (0.39)	43.65 (0.62)	20.40 (2.37)	11.49 (1.63)	84.78 (0.64)	35.79
+ LATTICE	**53.99 (0.65)**	**22.78 (1.26)**	43.57 (0.40)	37.89 (3.72)	**23.44 (4.01)**	**87.82 (1.47)**	**44.91**
+ learned rule, d=64	38.33 (0.84)	17.50 (0.79)	44.38 (0.77)	36.14 (2.01)	12.63 (1.75)	83.50 (1.19)	38.74
+ learned rule, d=128	40.66 (0.67)	22.13 (0.77)	44.97 (0.97)	36.70 (2.65)	15.07 (1.15)	84.66 (1.24)	40.70
+ learned rule, d=256	41.39 (0.66)	21.74 (1.50)	**45.14 (0.63)**	**41.03 (1.07)**	17.30 (1.46)	85.78 (0.82)	42.06
AIN	46.82 (0.89)	55.76 (1.19)	**59.55 (2.91)**	47.24 (2.41)	59.20 (1.92)	84.07 (2.46)	58.77
+ LATTICE	**56.72 (0.51)**	**67.25 (1.58)**	57.92 (2.49)	**60.50 (0.62)**	**76.49 (4.18)**	**92.89 (1.22)**	**68.63**
+ learned rule, d=64	51.36 (1.47)	58.01 (1.13)	55.27 (0.69)	59.55 (1.39)	69.32 (1.25)	90.07 (1.28)	63.93
+ learned rule, d=128	49.83 (1.05)	60.33 (1.17)	55.30 (0.71)	57.05 (3.77)	70.62 (2.01)	90.92 (0.92)	64.01
+ learned rule, d=256	50.80 (0.96)	57.25 (1.78)	56.30 (1.19)	53.12 (5.91)	71.52 (1.93)	91.75 (1.39)	63.46
ED	49.22 (0.47)	**69.99 (0.82)**	**68.92 (3.26)**	50.33 (2.88)	70.49 (1.37)	90.17 (2.43)	**66.52**
+ LATTICE	38.00 (0.77)	64.51 (1.26)	64.20 (3.58)	58.52 (3.56)	**77.02 (1.75)**	89.90 (1.41)	65.36
+ learned rule, d=64	48.08 (0.96)	48.83 (2.99)	57.50 (3.86)	58.77 (0.95)	74.21 (2.73)	**96.72 (1.08)**	64.02
+ learned rule, d=128	**49.34 (0.35)**	47.60 (1.99)	59.92 (2.57)	58.48 (2.60)	72.57 (2.12)	94.42 (1.81)	63.72
+ learned rule, d=256	49.07 (1.00)	47.31 (2.92)	53.26 (2.14)	**59.62 (1.93)**	69.63 (4.62)	93.08 (1.92)	61.99

E.3 Statistical Significance

Results vary considerably across seeds on several benchmarks (Table 1), so we test the main contrasts with paired tests. Runs are paired by benchmark and seed. The hard values, and hence the held-out test set, are fixed for each benchmark and identical across seeds; a seed fixes the train/validation partition and the initialization, and runs with the same seed share the same partition for every method, so pairing removes the variation due to the partition, though not the variation due to initialization. For each backbone we report, over the 30 (benchmark, seed) pairs, the mean paired difference, a 95% confidence interval obtained by bootstrap resampling of benchmarks (the level at which the claims generalize), the number of pairs in which the second method wins, and the p-value of a two-sided Wilcoxon signed-rank test with Holm correction over the contrasts of each backbone. As a robustness check, Table 12 repeats the accuracy tests with one pair per benchmark (seeds averaged); with six pairs no two-sided p-value can fall below 0.0312, so most contrasts are not significant at that level for lack of power. Table 13 applies the same tests to the P-score and to $\rho_{(\mathrm{I})}$. The two tests answer different questions: the Wilcoxon test pools the 30 (benchmark, seed) pairs, while the cluster

bootstrap asks whether the effect holds across benchmarks. Where they disagree (a significant p with a confidence interval that contains 0, as for the test P-score of ResNet-18 from Base to + LATTICE), the effect is consistent across seeds but not established across benchmarks.

Table 12: Paired comparisons on test accuracy. Each pair is one (benchmark, seed): the held-out test set is fixed per benchmark, and the same seed fixes the same train/validation partition for both methods, so the difference removes the variation shared by them. Δ is the mean paired difference in test accuracy (%); the 95% CI is a percentile bootstrap that resamples *benchmarks* (cluster bootstrap), which is the level the claim generalises over. p is a two-sided Wilcoxon signed-rank test, Holm-corrected over the nine contrasts within each backbone; bold marks $p < 0.05$ after correction. The last two columns repeat the test with one pair per benchmark (seeds averaged, $n = 6$), where the smallest attainable two-sided p is 0.0312.

	Per (benchmark, seed), $n = 30$				**Per benchmark**, $n = 6$	
Contrast	**Δ (%)**	**95% CI**	**Wins**	**p (Holm)**	**Wins**	p
ResNet-18						
Base → + LATTICE	**+9.12**	[+2.65, +15.87]	24/30	**0.0003**	5/6	0.0625
Base → + CRM	-1.99	[-26.55, +21.27]	14/30	0.8855	3/6	1.0000
Base → + CRM_{loss}	**+10.76**	[+0.04, +22.63]	21/30	**0.0035**	4/6	0.1562
Base → + CRM + LATTICE	**+15.30**	[-1.27, +31.58]	20/30	**0.0030**	4/6	0.1562
Base → + CRM_{loss} + LATTICE	**+18.07**	[+8.55, +28.60]	30/30	<**0.0001**	6/6	0.0312
+ CRM → + CRM + LATTICE	**+17.29**	[+4.54, +30.80]	25/30	**0.0003**	5/6	0.0938
+ CRM_{loss} → + CRM_{loss} + LATTICE	**+7.31**	[+3.14, +11.33]	25/30	**0.0001**	5/6	0.0625
+ LATTICE → + CRM + LATTICE	+6.17	[-5.53, +17.31]	17/30	0.0877	4/6	0.3125
+ LATTICE → + CRM_{loss} + LATTICE	**+8.95**	[+3.96, +15.20]	28/30	<**0.0001**	6/6	0.0312
AIN						
Base → + LATTICE	**+9.85**	[+4.15, +14.84]	25/30	**0.0004**	5/6	0.0625
Base → + CRM	-7.47	[-22.98, +8.08]	10/30	0.2450	2/6	0.5625
Base → + CRM_{loss}	**+9.56**	[+3.64, +15.22]	24/30	**0.0004**	5/6	0.0625
Base → + CRM + LATTICE	**+9.24**	[-0.35, +18.37]	23/30	**0.0072**	4/6	0.1562
Base → + CRM_{loss} + LATTICE	**+12.29**	[+8.15, +16.60]	29/30	<**0.0001**	6/6	0.0312
+ CRM → + CRM + LATTICE	**+16.71**	[+8.59, +23.70]	27/30	<**0.0001**	5/6	0.0625
+ CRM_{loss} → + CRM_{loss} + LATTICE	+2.73	[-2.60, +8.22]	16/30	0.4858	4/6	0.3125
+ LATTICE → + CRM + LATTICE	-0.61	[-11.01, +9.38]	16/30	0.6959	3/6	1.0000
+ LATTICE → + CRM_{loss} + LATTICE	+2.44	[-1.63, +7.23]	18/30	0.4858	3/6	0.4375
ED						
Base → + LATTICE	-1.16	[-7.07, +4.96]	12/30	0.9705	2/6	0.8438
Base → + CRM	-12.43	[-31.14, +5.72]	10/30	0.0769	2/6	0.4375
Base → + CRM_{loss}	+5.15	[-3.15, +13.29]	21/30	0.1062	5/6	0.2188
Base → + CRM + LATTICE	+4.26	[-5.22, +14.37]	18/30	0.8384	4/6	0.5625
Base → + CRM_{loss} + LATTICE	**+5.59**	[-0.33, +10.89]	20/30	**0.0319**	5/6	0.0938
+ CRM → + CRM + LATTICE	**+16.68**	[+7.09, +26.57]	26/30	**0.0002**	6/6	0.0312
+ CRM_{loss} → + CRM_{loss} + LATTICE	+0.45	[-4.96, +5.39]	14/30	0.9705	4/6	0.6875
+ LATTICE → + CRM + LATTICE	+5.42	[-4.81, +18.21]	14/30	0.9705	3/6	0.4375
+ LATTICE → + CRM_{loss} + LATTICE	**+6.76**	[-0.12, +15.09]	23/30	**0.0384**	5/6	0.0938

Table 13: Paired comparisons on the representation metrics, over the same (benchmark, seed) pairs as Table 12. Δ is the mean paired difference; the 95% CI is a cluster bootstrap over benchmarks; p is a two-sided Wilcoxon signed-rank test, Holm-corrected over the five contrasts within each backbone and metric. Bold marks $p < 0.05$ after correction regardless of the direction of the effect. Higher is better for P-score, lower for $\rho_{(\mathrm{I})}$, so a favourable effect is $\Delta > 0$ for the first two blocks and $\Delta < 0$ for the third.

Metric	Contrast	Δ	95% CI	$\Delta > 0$	p (Holm)
P-score (val)	*ResNet-18*: Base → + LATTICE	**+0.257**	[+0.158, +0.358]	30/30	**<0.0001**
	Base → + CRM	**-0.055**	[-0.094, -0.016]	4/30	**0.0001**
	Base → + CRM + LATTICE	**+0.253**	[+0.157, +0.353]	30/30	**<0.0001**
	+ CRM → + CRM + LATTICE	**+0.308**	[+0.233, +0.387]	30/30	**<0.0001**
	+ LATTICE → + CRM + LATTICE	**-0.003**	[-0.010, +0.001]	5/30	**0.0040**
	AIN: Base → + LATTICE	**+0.208**	[+0.133, +0.303]	30/30	**<0.0001**
	Base → + CRM	**+0.033**	[+0.009, +0.062]	23/30	**0.0005**
	Base → + CRM + LATTICE	**+0.203**	[+0.131, +0.297]	30/30	**<0.0001**
	+ CRM → + CRM + LATTICE	**+0.171**	[+0.084, +0.279]	30/30	**<0.0001**
	+ LATTICE → + CRM + LATTICE	**-0.004**	[-0.008, -0.002]	0/30	**<0.0001**
	ED: Base → + LATTICE	**+0.028**	[+0.001, +0.067]	27/30	**0.0003**
	Base → + CRM	**-0.028**	[-0.055, -0.006]	2/30	**<0.0001**
	Base → + CRM + LATTICE	**+0.048**	[+0.010, +0.089]	29/30	**<0.0001**
	+ CRM → + CRM + LATTICE	**+0.076**	[+0.018, +0.138]	30/30	**<0.0001**
	+ LATTICE → + CRM + LATTICE	+0.020	[-0.003, +0.063]	7/30	0.2096
P-score (test)	*ResNet-18*: Base → + LATTICE	**+0.089**	[-0.004, +0.194]	22/30	**0.0038**
	Base → + CRM	**-0.029**	[-0.055, -0.004]	10/30	**0.0038**
	Base → + CRM + LATTICE	**+0.119**	[+0.035, +0.209]	25/30	**0.0001**
	+ CRM → + CRM + LATTICE	**+0.148**	[+0.067, +0.230]	26/30	**<0.0001**
	+ LATTICE → + CRM + LATTICE	**+0.030**	[+0.003, +0.059]	21/30	**0.0037**
	AIN: Base → + LATTICE	**+0.092**	[+0.040, +0.146]	29/30	**<0.0001**
	Base → + CRM	**+0.023**	[+0.007, +0.042]	21/30	**0.0029**
	Base → + CRM + LATTICE	**+0.113**	[+0.057, +0.170]	30/30	**<0.0001**
	+ CRM → + CRM + LATTICE	**+0.090**	[+0.042, +0.144]	29/30	**<0.0001**
	+ LATTICE → + CRM + LATTICE	**+0.021**	[+0.010, +0.035]	26/30	**0.0002**
	ED: Base → + LATTICE	**+0.038**	[+0.007, +0.070]	23/30	**0.0018**
	Base → + CRM	+0.014	[-0.011, +0.047]	16/30	0.2096
	Base → + CRM + LATTICE	**+0.076**	[+0.034, +0.119]	28/30	**<0.0001**
	+ CRM → + CRM + LATTICE	**+0.062**	[+0.020, +0.109]	24/30	**0.0003**
	+ LATTICE → + CRM + LATTICE	**+0.039**	[-0.002, +0.093]	19/30	**0.0133**
$\rho_{(\mathrm{I})}$	*ResNet-18*: Base → + LATTICE	**-0.137**	[-0.235, -0.035]	6/30	**0.0003**
	Base → + CRM	+0.004	[-0.039, +0.050]	12/30	0.8531
	Base → + CRM + LATTICE	**-0.168**	[-0.251, -0.083]	2/30	**<0.0001**
	+ CRM → + CRM + LATTICE	**-0.172**	[-0.242, -0.099]	0/30	**<0.0001**
	+ LATTICE → + CRM + LATTICE	-0.030	[-0.076, +0.017]	10/30	0.0527
	AIN: Base → + LATTICE	**-0.228**	[-0.296, -0.164]	0/30	**<0.0001**
	Base → + CRM	**-0.068**	[-0.105, -0.033]	2/30	**<0.0001**
	Base → + CRM + LATTICE	**-0.257**	[-0.315, -0.208]	0/30	**<0.0001**
	+ CRM → + CRM + LATTICE	**-0.189**	[-0.224, -0.157]	0/30	**<0.0001**
	+ LATTICE → + CRM + LATTICE	**-0.029**	[-0.050, -0.008]	7/30	**0.0009**
	ED: Base → + LATTICE	**-0.212**	[-0.278, -0.161]	0/30	**<0.0001**
	Base → + CRM	**-0.083**	[-0.152, -0.022]	6/30	**0.0003**
	Base → + CRM + LATTICE	**-0.241**	[-0.317, -0.172]	0/30	**<0.0001**
	+ CRM → + CRM + LATTICE	**-0.158**	[-0.202, -0.112]	1/30	**<0.0001**
	+ LATTICE → + CRM + LATTICE	-0.029	[-0.095, +0.030]	11/30	0.1229

F QUALITATIVE RESULTS

F.1 REPRESENTATION DECODING

To visualize the structure encoded in the learned representations, we train a decoder that maps latent vectors back to the image space. The decoder is implemented as a 6-layer convolutional architecture based on transposed convolutions (deconvolutions), which progressively upsamples the latent representation into a full-resolution image (64×64). This design introduces an explicit spatial inductive bias, encouraging the reconstruction to respect local structure and natural image statistics.

The decoder is trained **post hoc**, with the representation model kept frozen. For each input image x, we first extract its latent representation $z = f_\theta(x)$ using the pretrained encoder. The decoder (d_γ) is then optimized to reconstruct the original image by minimizing a mean squared error (MSE) or cross-entropy (CE) objective.

Training is performed over the full dataset, ensuring that the decoder learns a global inverse mapping from the representation space to the pixel space without issues arising from the dataset's distributional properties (used to test compositional generalization).

Qualitative results are obtained by sampling images from the dataset, encoding them into representations, and passing these representations through the trained decoder to produce reconstructions.

F.2 Additional Examples of Imagined Representations

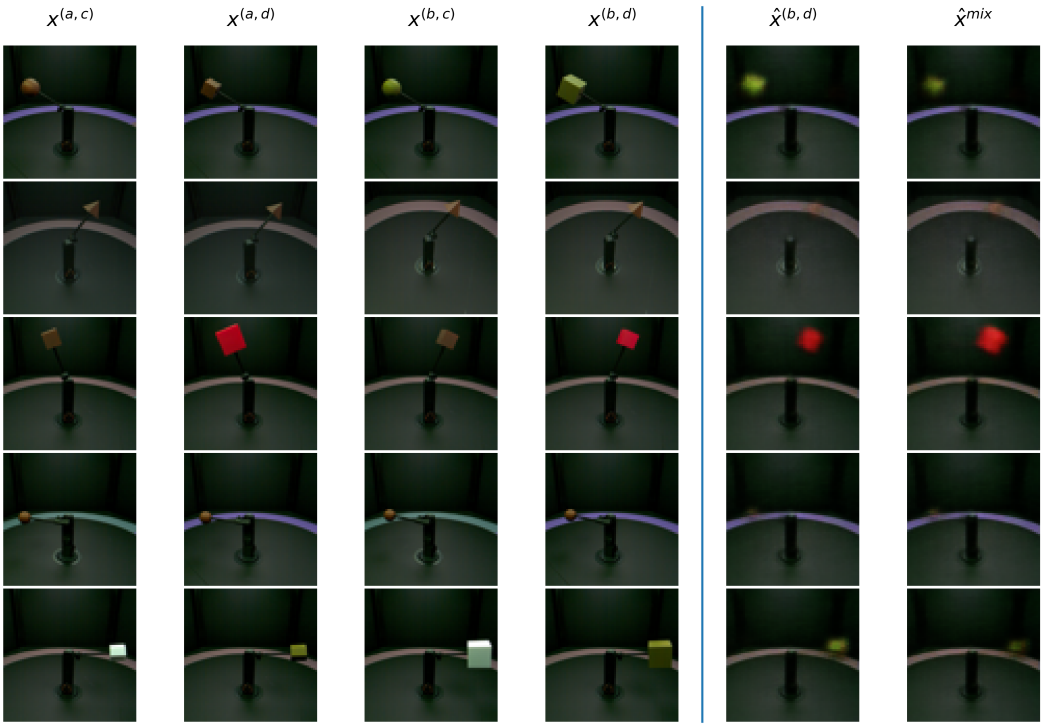

Figure 4: Additional qualitative visualization of the representations of AIN + LATTICE predicted by the analogy rule I_{ana} on MPI3D. In each test-set example, the first three columns show the input images $x^{(a,c)}$, $x^{(a,d)}$, and $x^{(b,c)}$, whose encoder representations $f_\theta(x^{(a,c)})$, $f_\theta(x^{(a,d)})$, and $f_\theta(x^{(b,c)})$ are given to the analogy rule I_{ana} to predict the representation of the held-out target composition $x^{(b,d)}$ shown in the fourth column. To the right of the vertical divider, $\hat{x}^{(b,d)}$ is obtained by decoding the true target representation $f_\theta(x^{(b,d)})$, while $\hat{x}^{mix}$ is obtained by decoding the representation predicted by I_{ana}, with the post-hoc decoder of Appendix F.1, trained on the frozen encoder over all compositions, including held-out ones, and used only for visualization. Although the decoded images are blurrier than the originals, $\hat{x}^{mix}$ preserves the main factors of $x^{(b,d)}$ and closely matches the reconstruction from its true encoder representation.

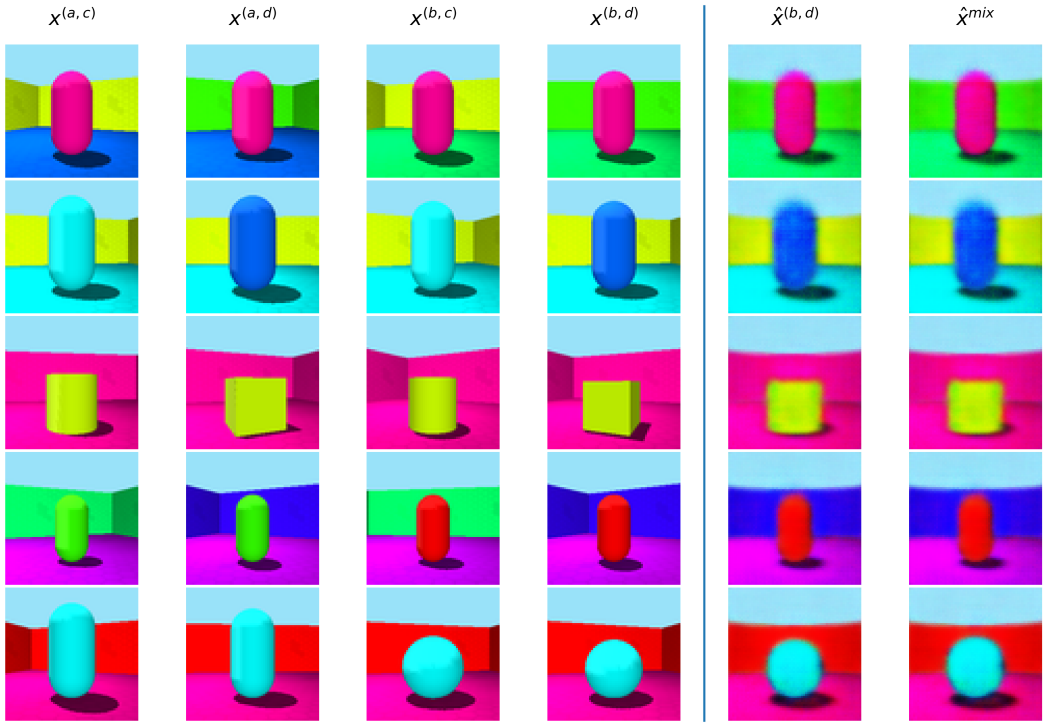

Figure 5: Qualitative visualization of the representations of AIN + LATTICE predicted by the analogy rule I_{ana} on Shapes3D. In each test-set example, the first three columns show the input images $x^{(a,c)}$, $x^{(a,d)}$, and $x^{(b,c)}$, whose encoder representations $f_\theta(x^{(a,c)})$, $f_\theta(x^{(a,d)})$, and $f_\theta(x^{(b,c)})$ are given to the analogy rule I_{ana} to predict the representation of the held-out target composition $x^{(b,d)}$ shown in the fourth column. To the right of the vertical divider, $\hat{x}^{(b,d)}$ is obtained by decoding the true target representation $f_\theta(x^{(b,d)})$, while $\hat{x}^{mix}$ is obtained by decoding the representation predicted by I_{ana}, decoded as in Figure 4. Although the decoded images are blurrier than the originals, $\hat{x}^{mix}$ preserves the main factors of $x^{(b,d)}$ and closely matches the reconstruction from its true encoder representation.

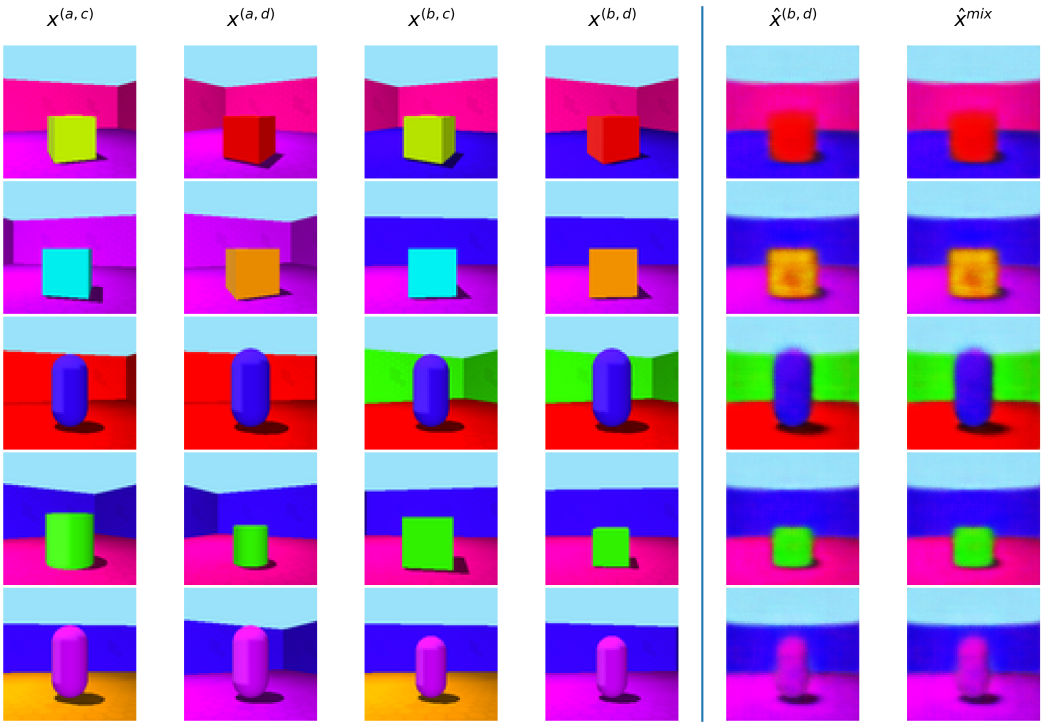

Figure 6: Additional qualitative visualization of the representations of AIN + LATTICE predicted by the analogy rule I_{ana} on Shapes3D. In each test-set example, the first three columns show the input images $x^{(a,c)}$, $x^{(a,d)}$, and $x^{(b,c)}$, whose encoder representations $f_\theta(x^{(a,c)})$, $f_\theta(x^{(a,d)})$, and $f_\theta(x^{(b,c)})$ are given to the analogy rule I_{ana} to predict the representation of the held-out target composition $x^{(b,d)}$ shown in the fourth column. To the right of the vertical divider, $\hat{x}^{(b,d)}$ is obtained by decoding the true target representation $f_\theta(x^{(b,d)})$, while $\hat{x}^{mix}$ is obtained by decoding the representation predicted by I_{ana}, decoded as in Figure 4. Although the decoded images are blurrier than the originals, $\hat{x}^{mix}$ preserves the main factors of $x^{(b,d)}$ and closely matches the reconstruction from its true encoder representation.

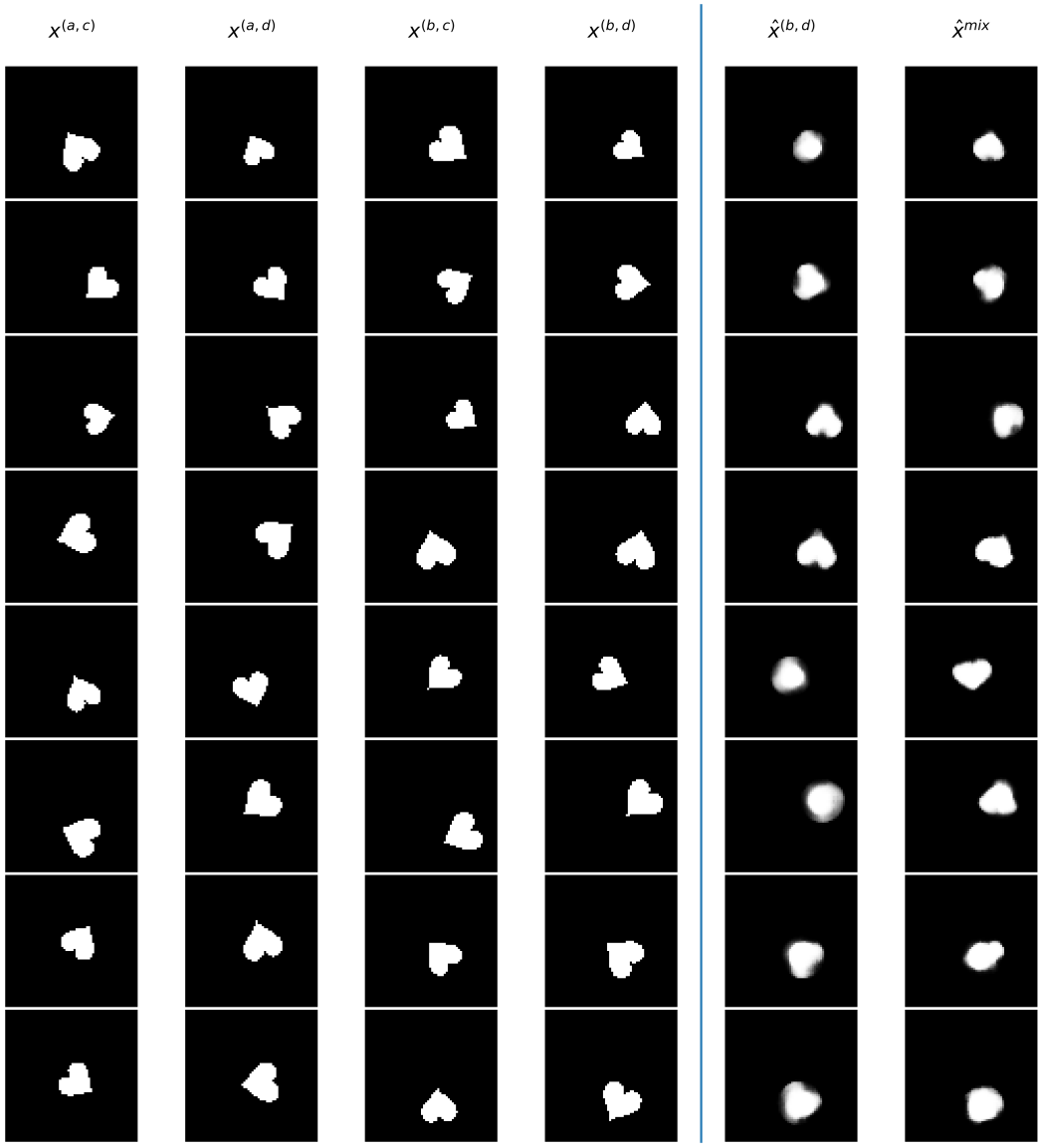

Figure 7: Qualitative visualization of the representations of AIN + LATTICE predicted by the analogy rule I_{ana} on dSprites. In each test-set example, the first three columns show the input images $x^{(a,c)}$, $x^{(a,d)}$, and $x^{(b,c)}$, whose encoder representations $f_\theta(x^{(a,c)})$, $f_\theta(x^{(a,d)})$, and $f_\theta(x^{(b,c)})$ are given to the analogy rule I_{ana} to predict the representation of the held-out target composition $x^{(b,d)}$ shown in the fourth column. To the right of the vertical divider, $\hat{x}^{(b,d)}$ is obtained by decoding the true target representation $f_\theta(x^{(b,d)})$, while $\hat{x}^{mix}$ is obtained by decoding the representation predicted by I_{ana}, decoded as in Figure 4. Although the decoded images are blurrier than the originals, $\hat{x}^{mix}$ preserves the main factors of $x^{(b,d)}$, although on dSprites it differs visibly from the reconstruction of the true encoder representation more often than on the other benchmarks.

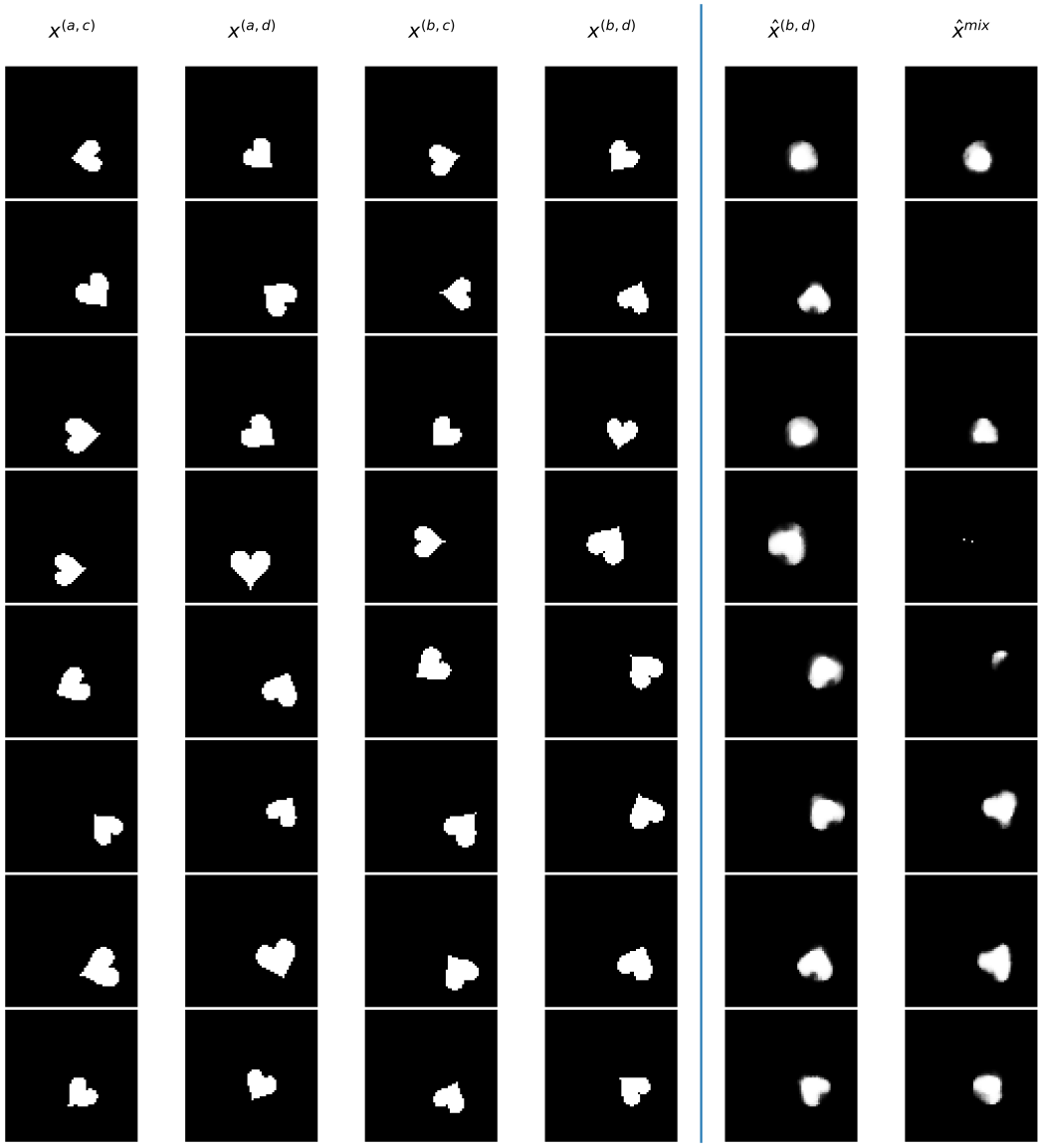

Figure 8: Additional qualitative visualization of the representations of AIN + LATTICE predicted by the analogy rule I_{ana} on dSprites. Sample of some failures of LATTICE. In each test-set example, the first three columns show the input images $x^{(a,c)}$, $x^{(a,d)}$, and $x^{(b,c)}$, whose encoder representations $f_\theta(x^{(a,c)})$, $f_\theta(x^{(a,d)})$, and $f_\theta(x^{(b,c)})$ are given to the analogy rule I_{ana} to predict the representation of the held-out target composition $x^{(b,d)}$ shown in the fourth column. To the right of the vertical divider, $\hat{x}^{(b,d)}$ is obtained by decoding the true target representation $f_\theta(x^{(b,d)})$, while $\hat{x}^{mix}$ is obtained by decoding the representation predicted by I_{ana}, decoded as in Figure 4.

G Representation Analysis

G.1 Representational Metrics

Beyond the P-score, which measures the parallel structure that imagination imposes (Section C.4), we characterize the representations that LATTICE induces with further measures of geometry and of representational complexity, to test whether LATTICE also simplifies the representation.

For geometry, we use Topographic Similarity (TopSim; Brighton & Kirby, 2006) and the Parallelism Score (P-score; Bernardi et al., 2020). TopSim measures the extent to which pairwise distances in latent space preserve pairwise distances in semantic space, with higher values indicating better topographic alignment. P-score measures whether attribute-specific displacement vectors remain aligned across contexts, i.e., whether the same semantic change induces a consistent direction in latent space. Higher P-score therefore indicates a more context-independent, parallel structure.

For representational complexity, we use three complementary measures. First, we estimate intrinsic dimensionality using Two-NN (Facco et al., 2017), which infers dimension from the ratio between the first and second nearest-neighbor distances. Second, we compute NC-90, defined as the number of singular components required to explain 90% of the variance. Third, we analyze the cumulative explained-variance curve as a function of the fraction of singular components used, and compute its area under the curve (SV-AUC). Higher SV-AUC indicates that variance is concentrated in fewer components, and therefore corresponds to lower representational complexity. Finally, we measure Hoyer sparsity (Hoyer, 2004), which quantifies how concentrated a representation is across units; higher values indicate a more sparse or selective code.

Table 3 in the main text reports the P-score averaged over benchmarks, and Tables 14 and 15 break it down by benchmark; the remaining metrics are shown as per-dataset training curves in the figures below. The improvement in compositional generalization is accompanied by two consistent representational changes: higher parallelism scores and lower Hoyer sparsity. The P-score increase holds on average for every backbone on both splits, and the curves show the same direction for Hoyer sparsity across datasets. In particular, the increase in P-score indicates that attribute-dependent transformations become more consistent across contexts, which is consistent with a more compositional representation, while the decrease in Hoyer sparsity indicates that this structure is implemented through a denser, more distributed code rather than through highly specialized units.

Per benchmark, the validation P-score nearly saturates for every model trained with LATTICE (at least 0.95 on all six benchmarks, with the single exception of ED + LATTICE on Cars3D, 0.85), so validation barely discriminates between them and the information lies in the held-out test split, where values still range from 0.63 to 0.99. Cars3D is the benchmark where the test P-score stays lowest for all models and where the differences between methods are largest.

Table 14: P-score on ID validation representations, per benchmark, averaged over 5 seeds with the standard error of the mean in parentheses. Higher is better; the best value per column is in bold, ties included. All values are computed on the checkpoint selected by validation accuracy, on the same runs as Table 1. CRM and CRM_{loss} share the trained encoder and differ only in the decision rule, so they have identical P-score and appear as one row. Shaded rows use LATTICE (ours).

Model	Cars3D	dSprites	MPI3D	CLEVR	I-RAVEN	Shapes3D	Avg
ResNet-18	0.534 (0.007)	0.803 (0.003)	0.521 (0.006)	0.868 (0.004)	0.857 (0.002)	0.808 (0.003)	0.732
ResNet-18 + CRM	0.545 (0.004)	0.724 (0.003)	0.512 (0.007)	0.764 (0.004)	0.829 (0.005)	0.687 (0.007)	0.677
ResNet-18 + LATTICE	0.972 (0.003)	0.999 (0.000)	0.960 (0.003)	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**	0.988
ResNet-18 + CRM + LATTICE	0.955 (0.004)	0.996 (0.000)	0.961 (0.003)	**1.000 (0.000)**	0.999 (0.000)	0.999 (0.001)	0.985
AIN	0.715 (0.005)	0.793 (0.010)	0.563 (0.008)	0.880 (0.014)	0.896 (0.008)	0.853 (0.004)	0.784
AIN + CRM	0.724 (0.010)	0.820 (0.008)	0.571 (0.016)	0.917 (0.015)	0.913 (0.012)	0.951 (0.003)	0.816
AIN + LATTICE	0.966 (0.003)	**1.000 (0.000)**	0.982 (0.002)	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**	0.991
AIN + CRM + LATTICE	0.961 (0.004)	0.988 (0.000)	0.977 (0.002)	0.999 (0.000)	0.999 (0.000)	0.998 (0.001)	0.987
ED	0.854 (0.008)	0.962 (0.002)	0.875 (0.008)	0.988 (0.002)	0.997 (0.000)	0.995 (0.003)	0.945
ED + CRM	0.797 (0.007)	0.957 (0.003)	0.794 (0.005)	0.981 (0.002)	0.992 (0.001)	0.982 (0.000)	0.917
ED + LATTICE	0.847 (0.010)	0.999 (0.000)	**0.994 (0.002)**	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**	0.973
ED + CRM + LATTICE	**0.973 (0.004)**	0.996 (0.000)	0.991 (0.001)	**1.000 (0.000)**	0.998 (0.001)	**1.000 (0.000)**	**0.993**

Table 15: P-score on held-out test representations, per benchmark, averaged over 5 seeds with the standard error of the mean in parentheses. Higher is better; the best value per column is in bold, ties included. All values are computed on the checkpoint selected by validation accuracy, on the same runs as Table 1. CRM and CRM_{loss} share the trained encoder and differ only in the decision rule, so they have identical P-score and appear as one row. Shaded rows use LATTICE (ours).

Model	Cars3D	dSprites	MPI3D	CLEVR	I-RAVEN	Shapes3D	Avg
ResNet-18	0.456 (0.004)	0.904 (0.004)	0.468 (0.004)	0.762 (0.004)	0.863 (0.004)	0.875 (0.005)	0.721
ResNet-18 + CRM	0.455 (0.007)	0.839 (0.005)	0.465 (0.008)	0.703 (0.010)	0.876 (0.005)	0.816 (0.005)	0.692
ResNet-18 + LATTICE	0.759 (0.004)	0.902 (0.002)	0.646 (0.008)	0.799 (0.016)	0.789 (0.008)	0.967 (0.008)	0.810
ResNet-18 + CRM + LATTICE	0.762 (0.004)	0.900 (0.004)	0.672 (0.007)	0.873 (0.014)	0.866 (0.004)	0.967 (0.007)	0.840
AIN	0.581 (0.006)	0.971 (0.002)	0.590 (0.011)	0.786 (0.021)	0.892 (0.008)	0.896 (0.007)	0.786
AIN + CRM	0.592 (0.006)	0.970 (0.002)	0.631 (0.017)	0.828 (0.011)	0.907 (0.004)	0.926 (0.002)	0.809
AIN + LATTICE	0.759 (0.004)	0.977 (0.003)	0.756 (0.010)	0.893 (0.011)	0.921 (0.017)	0.964 (0.011)	0.878
AIN + CRM + LATTICE	0.778 (0.003)	0.981 (0.004)	0.793 (0.014)	**0.911 (0.008)**	**0.953 (0.006)**	0.979 (0.004)	0.899
ED	0.641 (0.005)	0.977 (0.001)	0.795 (0.010)	0.810 (0.007)	0.831 (0.012)	0.949 (0.014)	0.834
ED + CRM	0.637 (0.004)	0.979 (0.001)	0.789 (0.008)	0.815 (0.038)	0.918 (0.008)	0.951 (0.003)	0.848
ED + LATTICE	0.631 (0.005)	**0.994 (0.000)**	0.889 (0.008)	0.862 (0.018)	0.900 (0.016)	0.955 (0.013)	0.872
ED + CRM + LATTICE	**0.793 (0.005)**	0.976 (0.001)	**0.901 (0.003)**	0.857 (0.011)	0.950 (0.006)	**0.985 (0.002)**	**0.910**

The picture for representational complexity is mixed. Two-NN decreases on average for every backbone, but only on three to five of the six datasets, and SV-AUC changes by less than 0.02 throughout, which is too small to support a claim of reduced complexity. NC-90 decreases markedly for ResNet-18 and ED (up to -15.9 components) but increases for AIN. Taken together, LATTICE reliably reorganizes the geometry toward a context-independent, parallel structure and toward a denser code, whereas its effect on the effective dimensionality of the representation is architecture-dependent rather than systematic.

G.2 IMAGINATION GAP

We report the imagination gap, term (I) of Theorem 2, as the normalized ratio $\rho_{(\mathrm{I})}$ of Algorithm 1, broken down by the number $j = j(y) = |T(y)|$ of hard values of the test configuration (Section 3.2): averaged over the six benchmarks (Table 16) and on MPI3D alone, the only benchmark covering $j = 2, \ldots, 6$ (Table 17).

Table 16: Normalized imagination gap $\rho_{(\mathrm{I})}$ by the number j of hard attribute values, averaged over the six benchmarks and 5 seeds, with the standard error of the mean over the n runs of each column in parentheses. For a test configuration y, $\rho_{(\mathrm{I})}$ is the residual of predicting $\varphi(y)$ by linear analogy from training configurations, normalized by the residual of the anchor mean: 0 means the unseen combination is exactly the superposition of its parts, 1 means imagining it is no better than the average of its anchors, and > 1 means the imagined representation is farther from the true one than the anchor mean, which the theory does not exclude, since the imagination gap is not controlled by any training-split objective. Lower is better; best per column in bold. Columns do not average over the same benchmarks, since the largest j is set by the number of split attributes: $j = 2, 3$ cover all six ($n = 30$), $j = 4$ four of them ($n = 20$), $j = 5$ MPI3D and Shapes3D ($n = 10$), and $j = 6$ MPI3D alone ($n = 5$). CRM and $\mathrm{CRM}_{\mathrm{loss}}$ share the trained encoder and differ only in the decision rule, so they have identical $\rho_{(\mathrm{I})}$ and appear as one row. Shaded rows use LATTICE (ours).

Model	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$	**All** j
ResNet-18	0.538 (0.022)	0.808 (0.033)	1.061 (0.055)	1.021 (0.062)	1.507 (0.021)	0.634 (0.022)
ResNet-18 + CRM	0.550 (0.020)	0.803 (0.036)	1.051 (0.068)	0.974 (0.056)	1.404 (0.037)	0.638 (0.022)
ResNet-18 + LATTICE	0.406 (0.030)	0.676 (0.042)	0.846 (0.058)	0.864 (0.097)	1.309 (0.045)	0.497 (0.035)
ResNet-18 + CRM + LATTICE	0.374 (0.024)	0.633 (0.039)	0.878 (0.057)	0.843 (0.081)	1.173 (0.025)	0.466 (0.030)
AIN	0.431 (0.021)	0.646 (0.031)	0.867 (0.040)	0.888 (0.058)	1.089 (0.017)	0.508 (0.022)
AIN + CRM	0.370 (0.018)	0.561 (0.032)	0.775 (0.056)	0.652 (0.065)	0.989 (0.030)	0.440 (0.021)
AIN + LATTICE	0.193 (0.018)	0.435 (0.037)	0.723 (0.068)	0.636 (0.092)	1.044 (0.076)	0.281 (0.023)
AIN + CRM + LATTICE	0.188 (0.016)	0.364 (0.031)	0.553 (0.052)	0.527 (0.089)	0.921 (0.068)	0.251 (0.020)
ED	0.358 (0.033)	0.638 (0.046)	0.854 (0.077)	0.675 (0.040)	0.801 (0.086)	0.454 (0.034)
ED + CRM	0.303 (0.019)	0.501 (0.035)	0.667 (0.058)	0.522 (0.048)	0.675 (0.103)	0.370 (0.022)
ED + LATTICE	0.174 (0.025)	0.369 (0.043)	0.537 (0.081)	0.313 (0.034)	0.300 (0.060)	0.242 (0.028)
ED + CRM + LATTICE	**0.154 (0.018)**	**0.331 (0.043)**	**0.508 (0.083)**	**0.228 (0.015)**	**0.208 (0.008)**	**0.212 (0.024)**

Table 17: Normalized imagination gap $\rho_{(\mathrm{I})}$ on MPI3D, averaged over 5 seeds, with standard error of the mean in parentheses. MPI3D is the only benchmark with $j = 2 \ldots 6$, so here the columns are comparable to each other and isolate the effect of j from the choice of benchmark. Lower is better; best per column in bold. Every row grows monotonically in j except ED + LATTICE, which flattens between $j = 5$ and $j = 6$ ($0.305 \to 0.300$). Shaded rows use LATTICE (ours).

Model	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$	**All** j
ResNet-18	0.485 (0.006)	0.749 (0.008)	1.006 (0.009)	1.202 (0.011)	1.507 (0.021)	0.611 (0.006)
ResNet-18 + CRM	0.444 (0.004)	0.674 (0.008)	0.926 (0.015)	1.135 (0.023)	1.404 (0.037)	0.557 (0.005)
ResNet-18 + LATTICE	0.441 (0.008)	0.750 (0.015)	0.984 (0.022)	1.147 (0.030)	1.309 (0.045)	0.584 (0.010)
ResNet-18 + CRM + LATTICE	0.372 (0.007)	0.655 (0.019)	0.896 (0.024)	1.061 (0.025)	1.173 (0.025)	0.508 (0.012)
AIN	0.409 (0.004)	0.611 (0.006)	0.803 (0.011)	0.940 (0.015)	1.089 (0.017)	0.505 (0.004)
AIN + CRM	0.370 (0.009)	0.540 (0.017)	0.709 (0.025)	0.842 (0.027)	0.989 (0.030)	0.453 (0.013)
AIN + LATTICE	0.267 (0.017)	0.499 (0.033)	0.731 (0.048)	0.895 (0.060)	1.044 (0.076)	0.379 (0.024)
AIN + CRM + LATTICE	0.227 (0.016)	0.414 (0.029)	0.618 (0.034)	0.783 (0.040)	0.921 (0.068)	0.320 (0.021)
ED	0.249 (0.016)	0.423 (0.035)	0.585 (0.052)	0.698 (0.068)	0.801 (0.086)	0.330 (0.024)
ED + CRM	0.249 (0.023)	0.387 (0.045)	0.507 (0.068)	0.592 (0.084)	0.675 (0.103)	0.313 (0.033)
ED + LATTICE	0.093 (0.024)	0.182 (0.047)	0.261 (0.064)	0.305 (0.066)	0.300 (0.060)	0.135 (0.034)
ED + CRM + LATTICE	**0.054 (0.002)**	**0.099 (0.004)**	**0.150 (0.006)**	**0.195 (0.008)**	**0.208 (0.008)**	**0.077 (0.003)**

G.3 CORRELATION BETWEEN REPRESENTATIONAL METRICS AND TEST ACCURACY

We also studied the correlation between representational metrics and test accuracy, to assess whether any of them could support model selection beyond ID performance. Metrics are computed on the ID

validation split of the checkpoint selected by validation accuracy, so they are available at selection time. The three architectures differ both in accuracy and in the typical range of these metrics, so pooling them conflates the architecture with the quality of its representations. We therefore compute Spearman correlations within each dataset and architecture, over its 4 training methods and 5 seeds (20 runs), and average them over the six datasets (Table 18).

Four metrics keep the same sign on all three architectures: P-score (positive), and TopSim, Hoyer sparsity and $\rho_{(\mathrm{I})}$ (negative). P-score is the strongest and most stable, and it is weakest on ED, where its values are already high and compressed. NC90 and SV-AUC change sign across architectures (SV-AUC is $+0.380$ on ResNet-18 but -0.178 on ED), so any statement about them has to be made per architecture; pooled over all architectures, SV-AUC appears strongly predictive, which suggests that it mostly separates architectures of different accuracy rather than tracking generalization within one. $\rho_{(\mathrm{I})}$ has a magnitude comparable to P-score, but it is measured on the test split and needs test labels to group configurations, so it cannot be used for model selection.

We tried using P-score, SV-AUC and Hoyer Sparsity together with ID performance for model selection; none of these combinations yielded better results.

Table 18: Spearman correlation between representational metrics and test accuracy, computed within each dataset and architecture (4 training methods $\times$ 5 seeds $=$ 20 runs) and averaged over the six datasets. Metrics are measured on the ID validation split, except $\rho_{(\mathrm{I})}$, which is measured on the test split. The last column averages the three architectures.

Metric	ResNet-18	AIN	ED	Mean
P-score	$+0.526$	$+0.555$	$+0.391$	$\mathbf{+0.490}$
TopSim	-0.025	-0.236	-0.327	$\mathbf{-0.196}$
TwoNN-ID	-0.165	-0.044	$+0.030$	$\mathbf{-0.060}$
NC90	-0.359	$+0.310$	$+0.193$	$\mathbf{+0.048}$
SV-AUC	$+0.380$	-0.032	-0.178	$\mathbf{+0.057}$
Hoyer	-0.446	-0.344	-0.132	$\mathbf{-0.307}$
$\rho_{(\mathrm{I})}$ **(test split)**	-0.479	-0.532	-0.376	$\mathbf{-0.462}$

G.4 Training Dynamics of Representational Metrics

We further analyze how these representational properties evolve during training. For each epoch, we evaluate six metrics on the validation set: Two-NN, TopSim, NC-90, SV-AUC, Hoyer Sparsity, and P-score. This allows us to track whether the geometric and complexity-related changes observed in the final representations emerge gradually during training, and whether they correlate with improvements in compositional generalization.

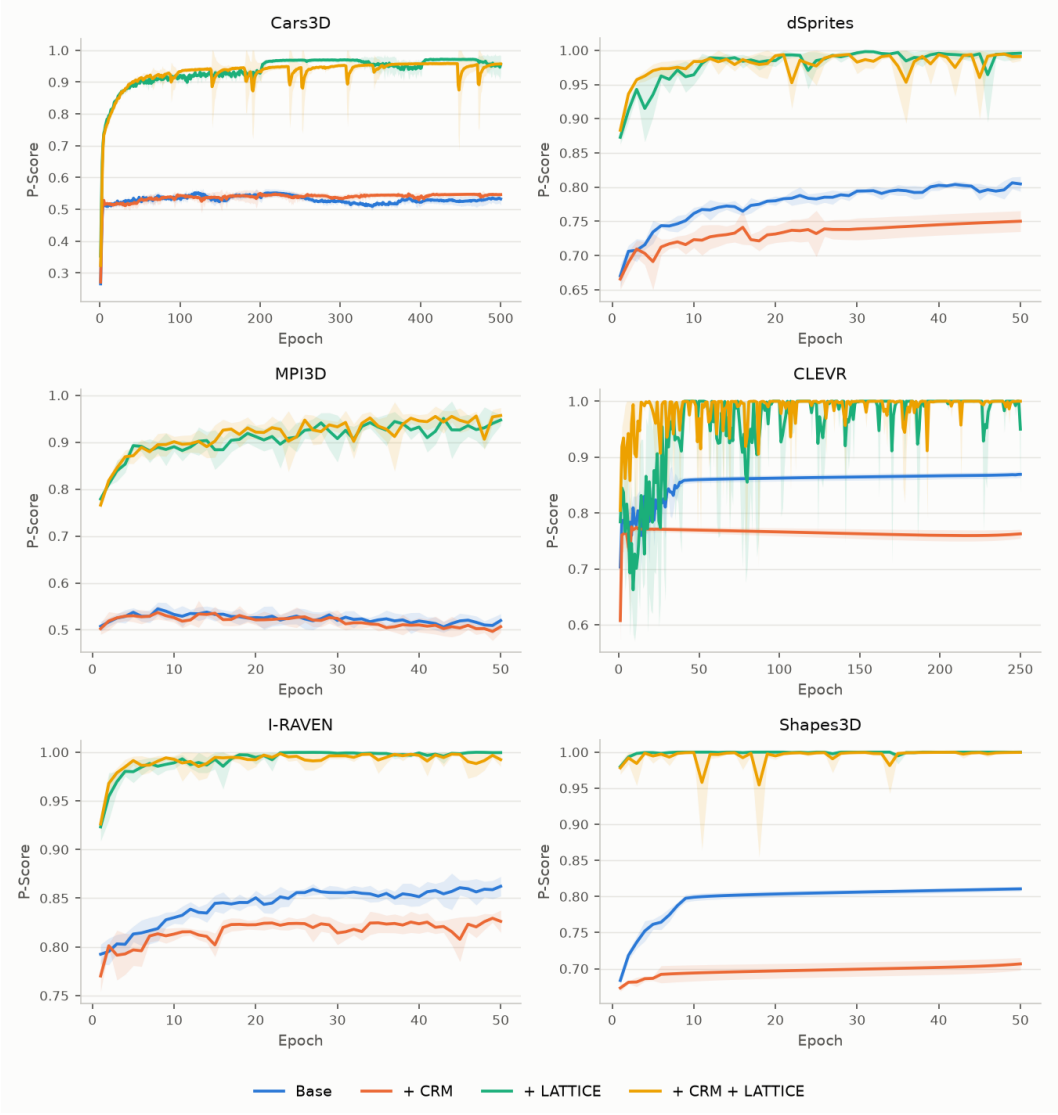

Figure 9: P-score over training epochs for ResNet-18 and versions of ResNet-18 augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

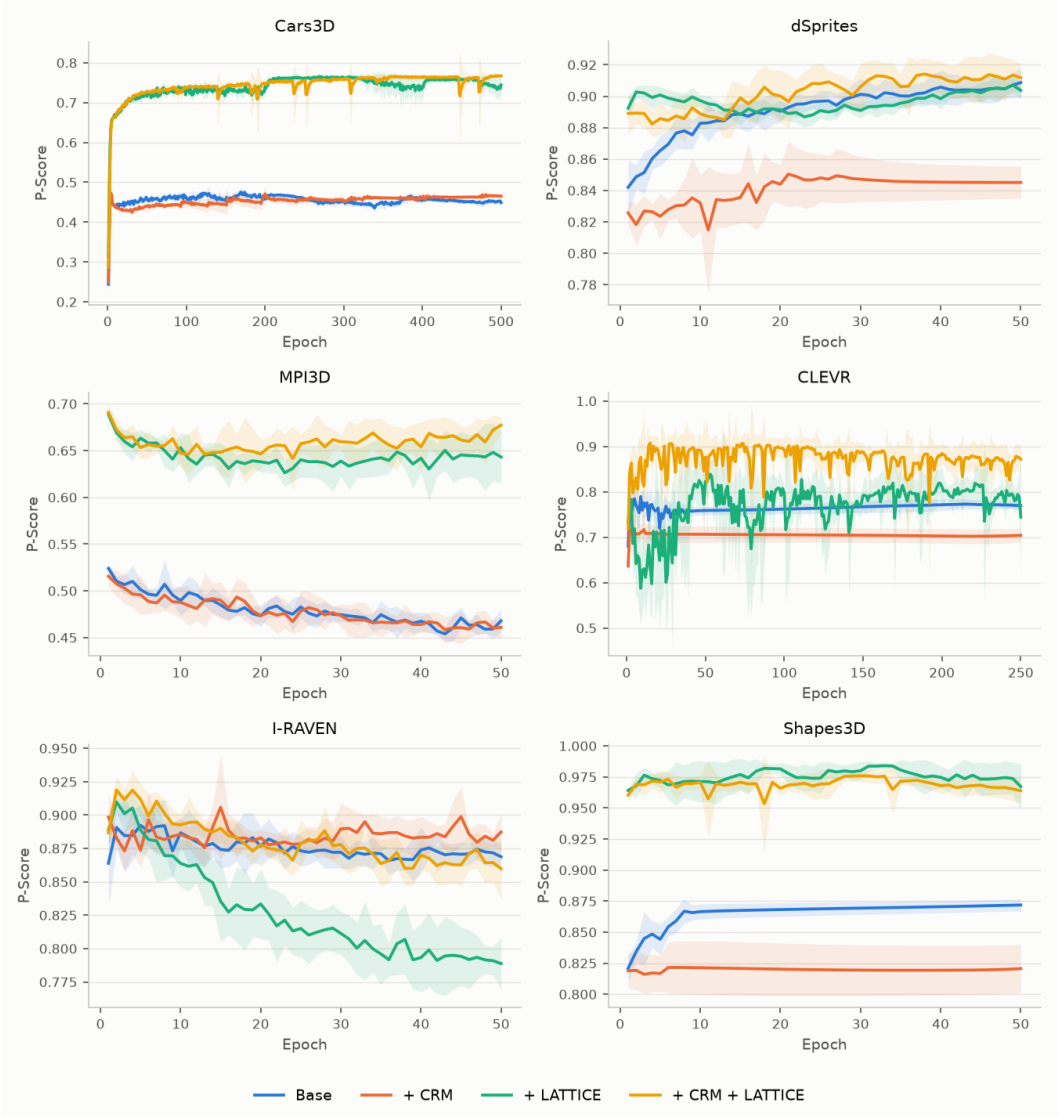

Figure 10: P-score over training epochs for ResNet-18 and versions of ResNet-18 augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

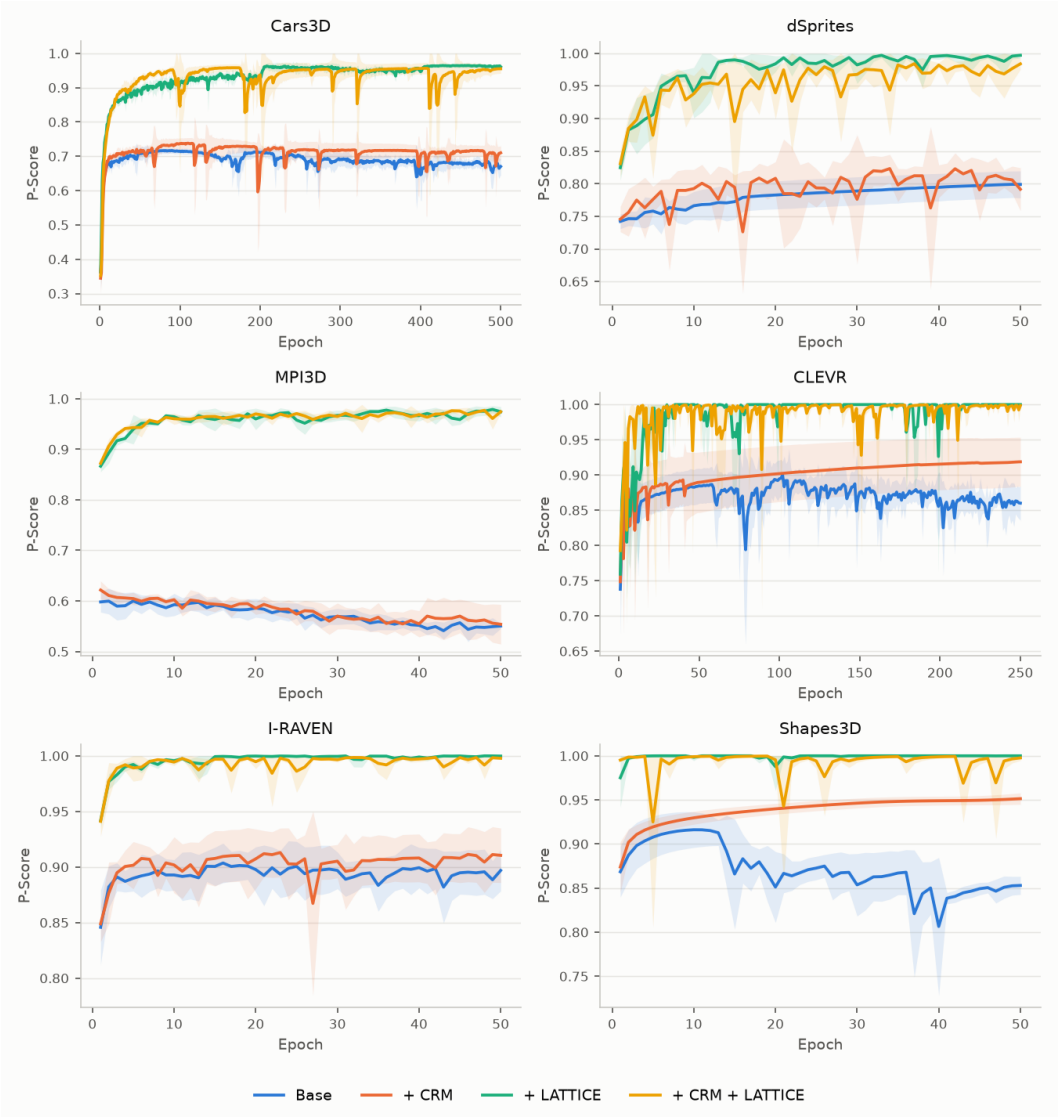

Figure 11: P-score over training epochs for AIN and versions of AIN augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

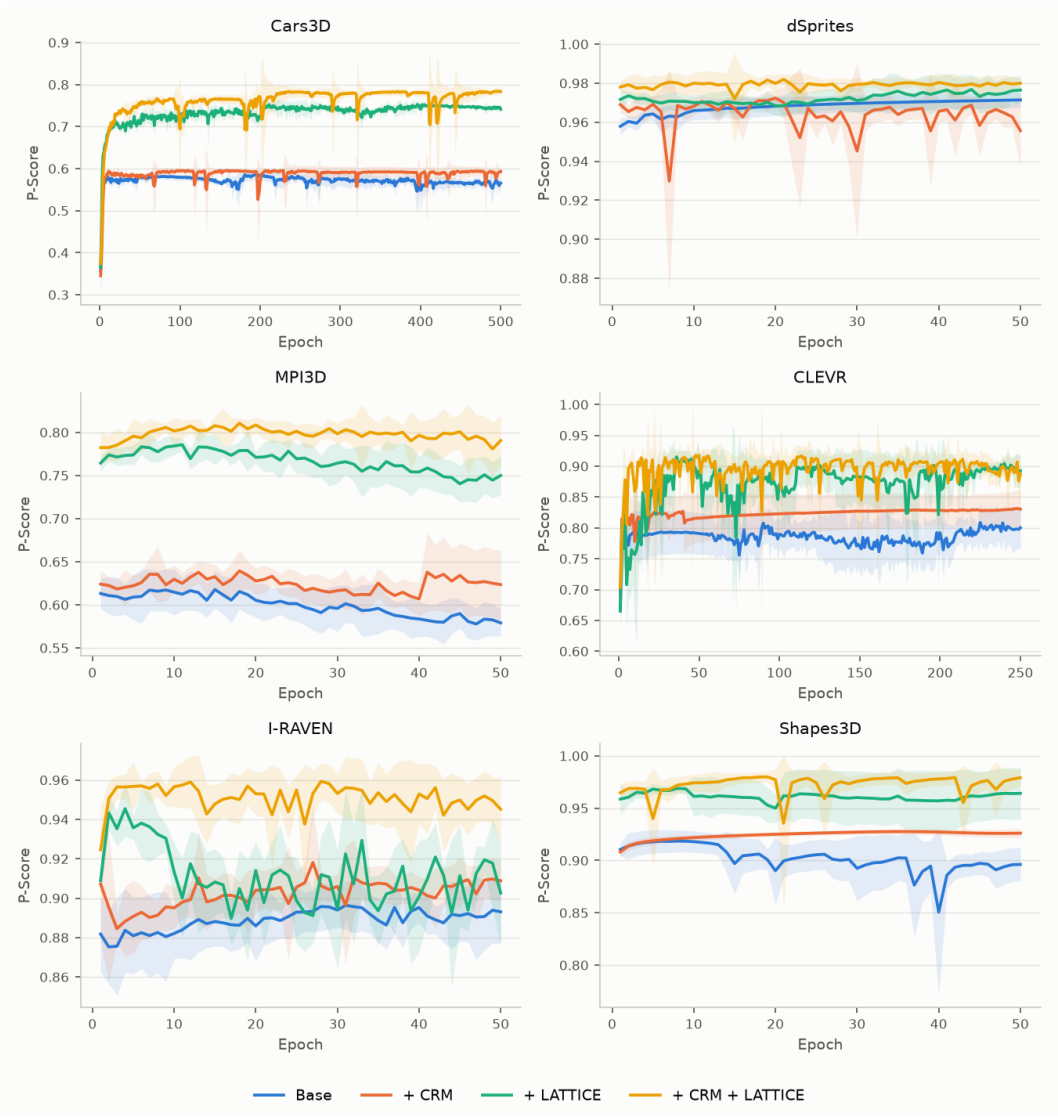

Figure 12: P-score over training epochs for AIN and versions of AIN augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

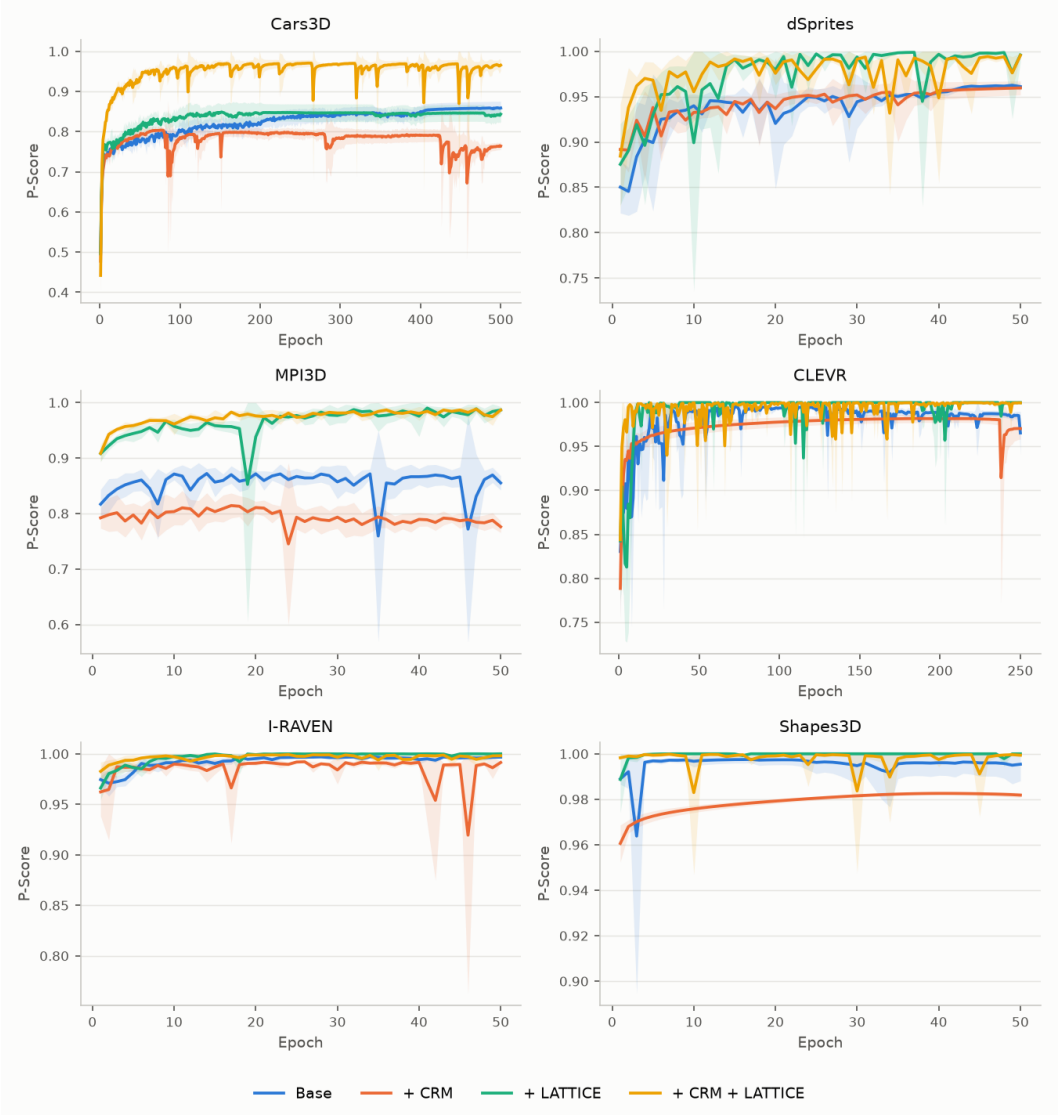

Figure 13: P-score over training epochs for ED and versions of ED augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

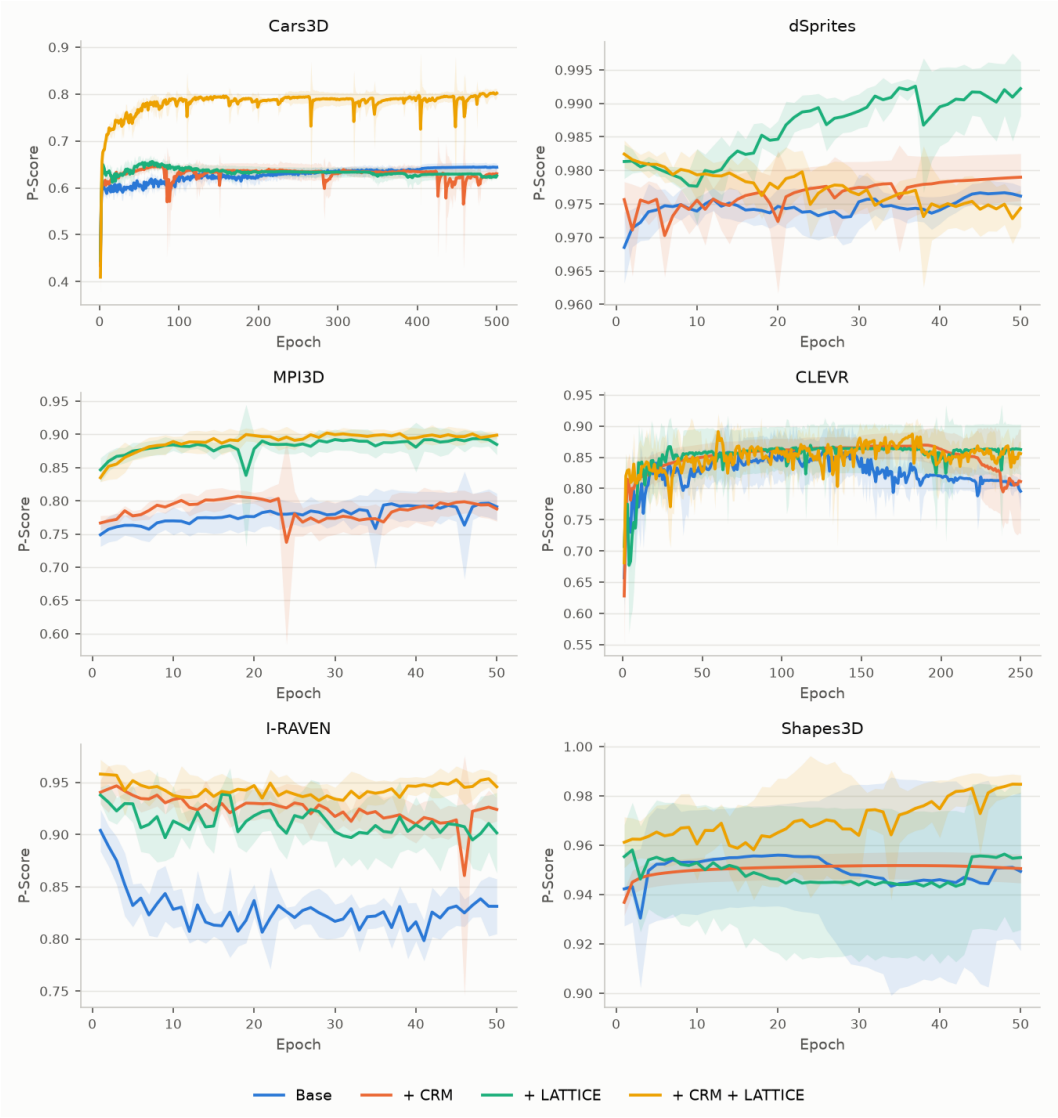

Figure 14: P-score over training epochs for ED and versions of ED augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

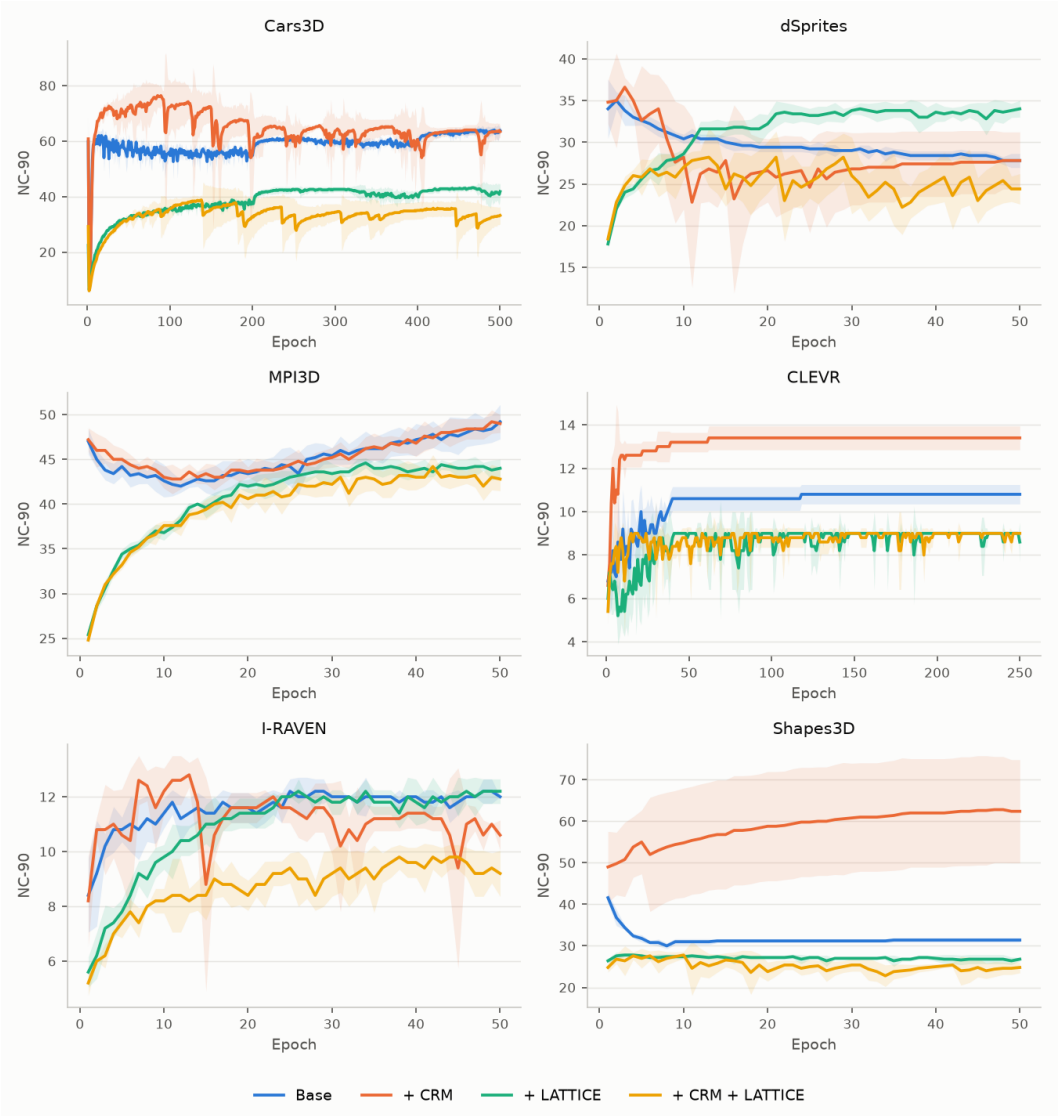

Figure 15: NC-90 over training epochs for ResNet-18 and versions of ResNet-18 augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

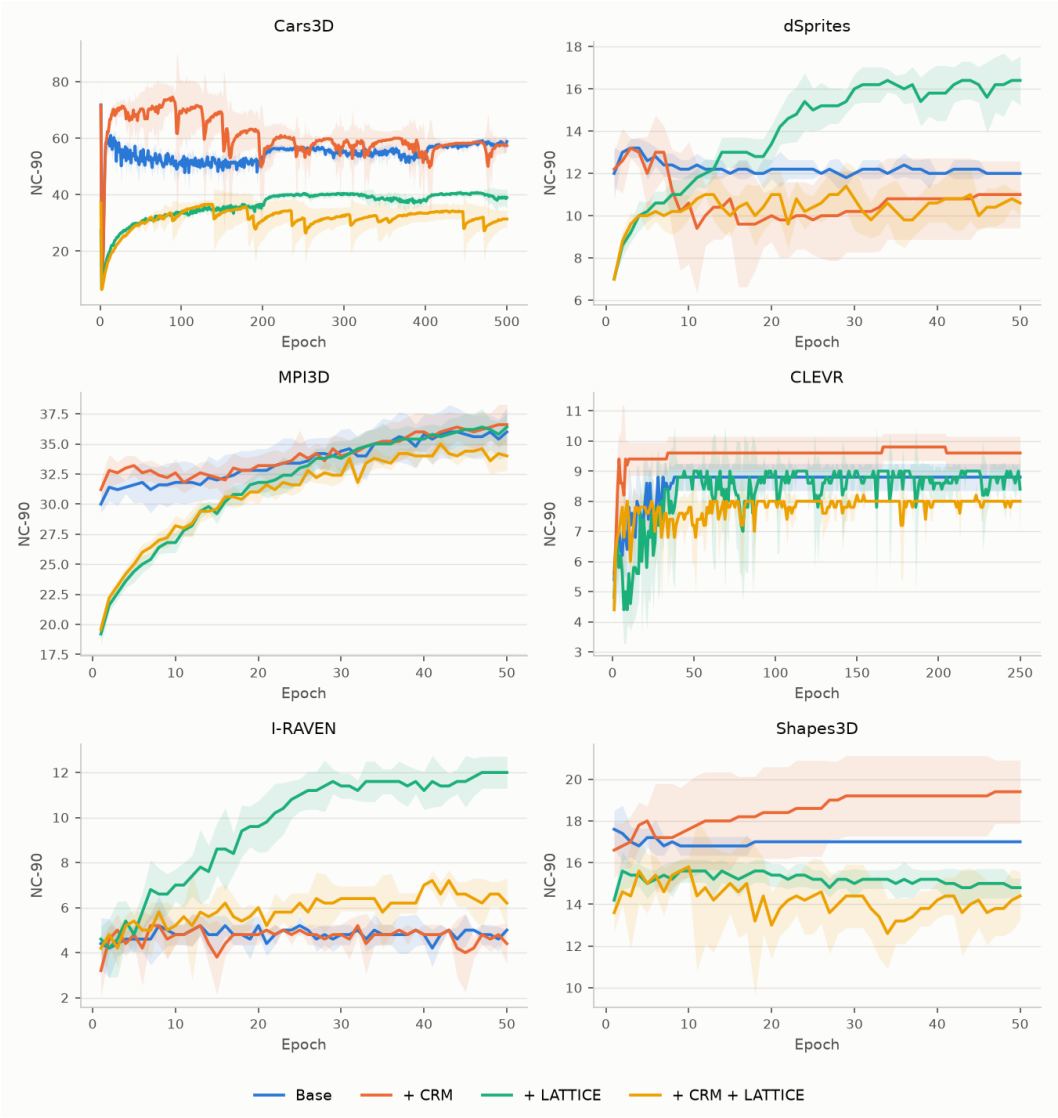

Figure 16: NC-90 over training epochs for ResNet-18 and versions of ResNet-18 augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

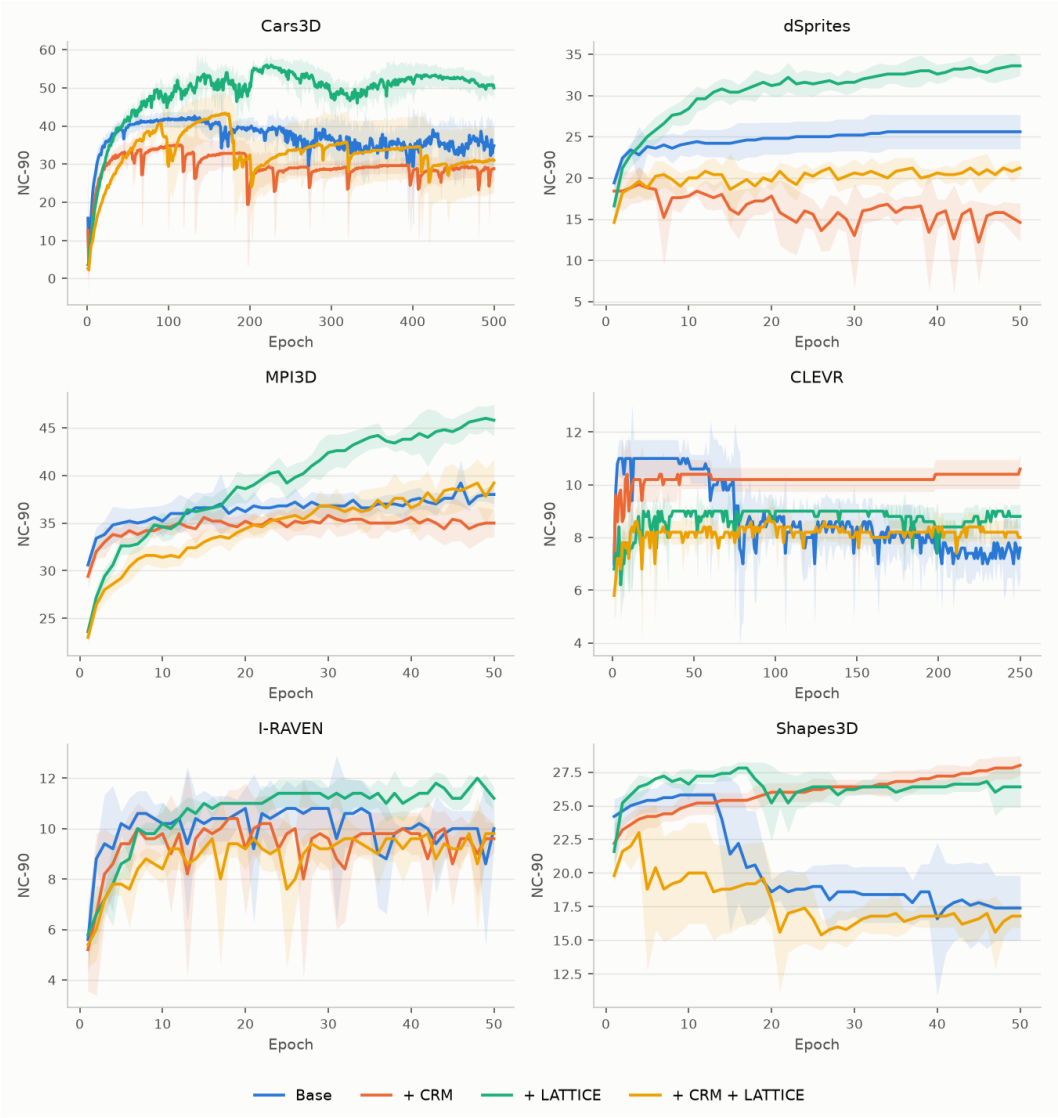

Figure 17: NC-90 over training epochs for AIN and versions of AIN augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

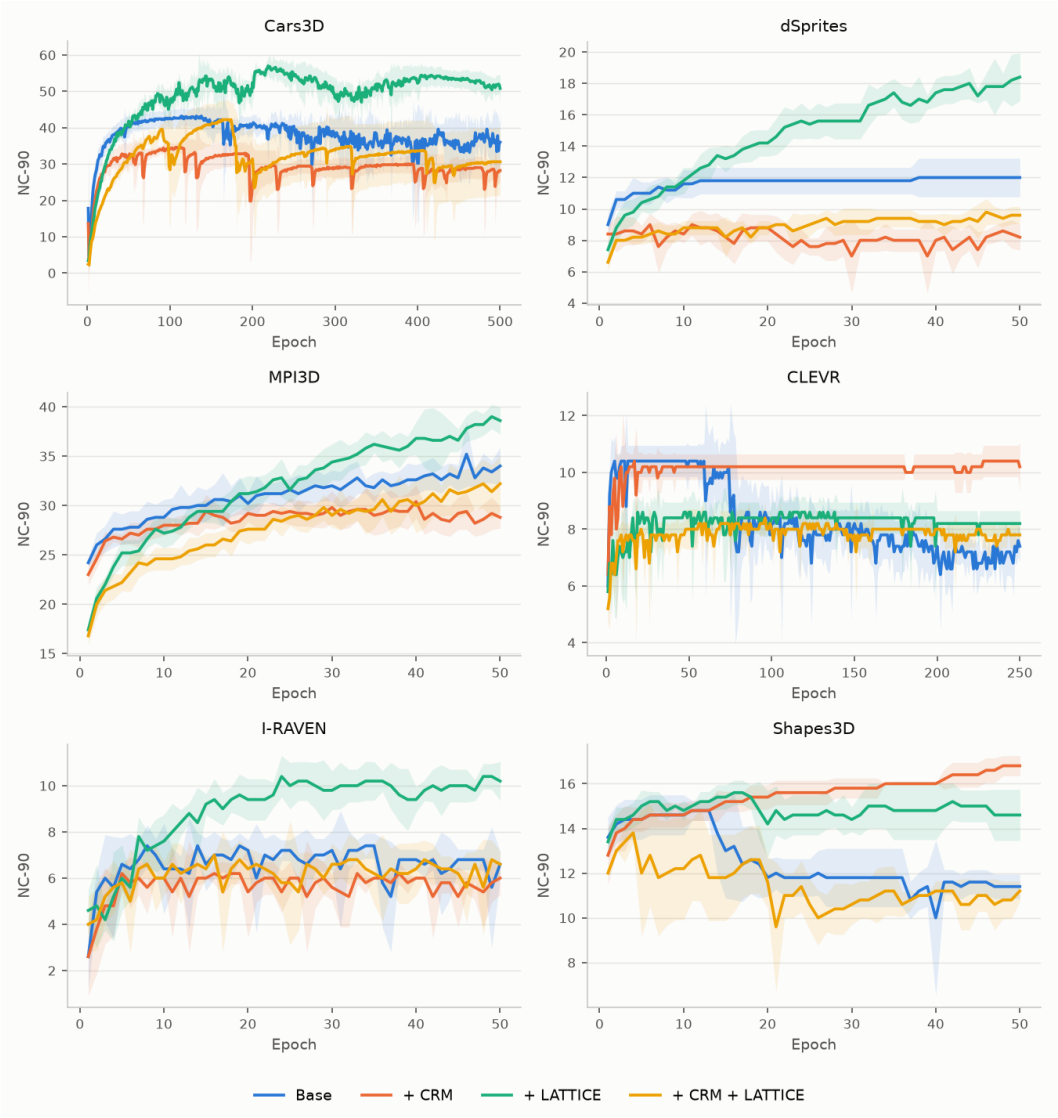

Figure 18: NC-90 over training epochs for AIN and versions of AIN augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

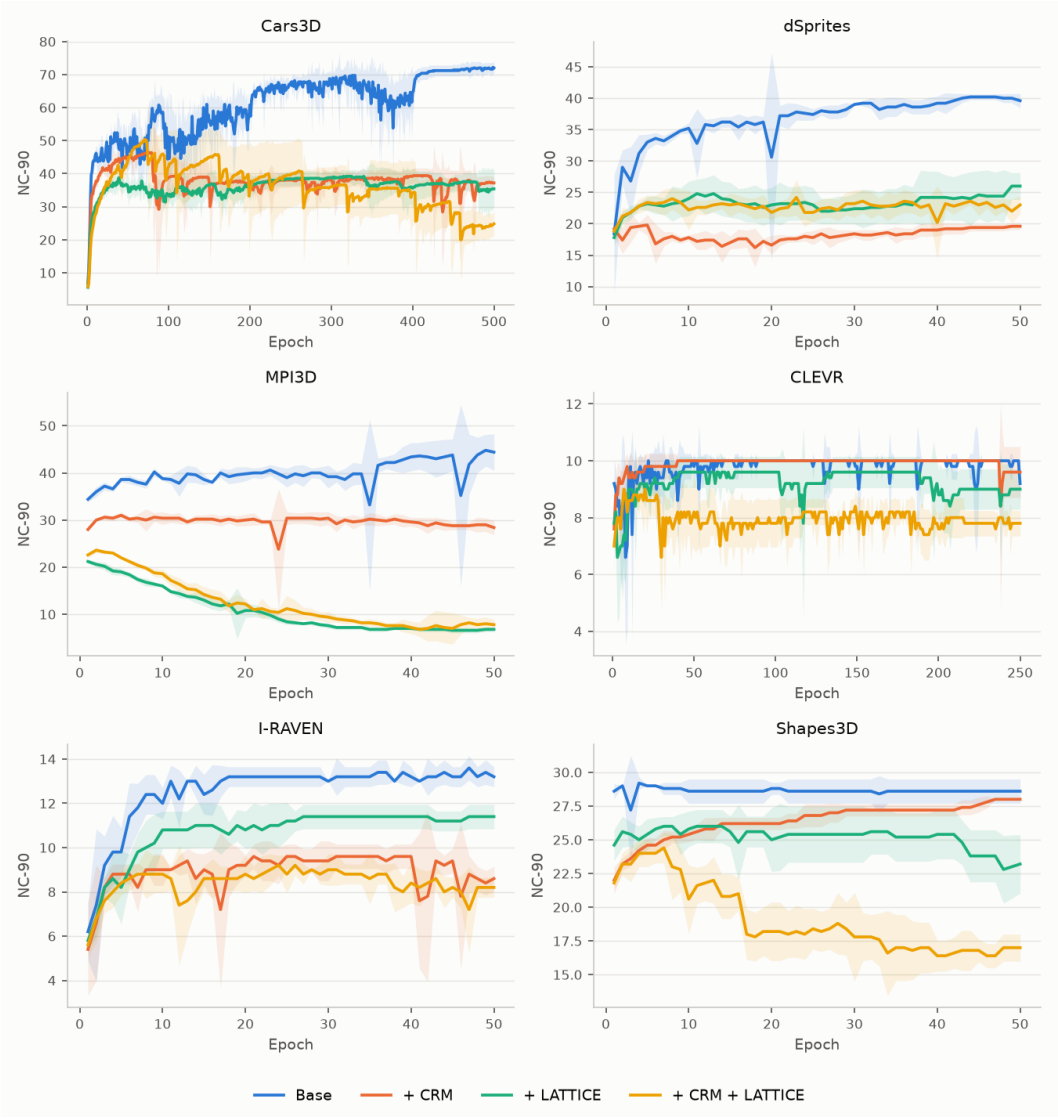

Figure 19: NC-90 over training epochs for ED and versions of ED augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

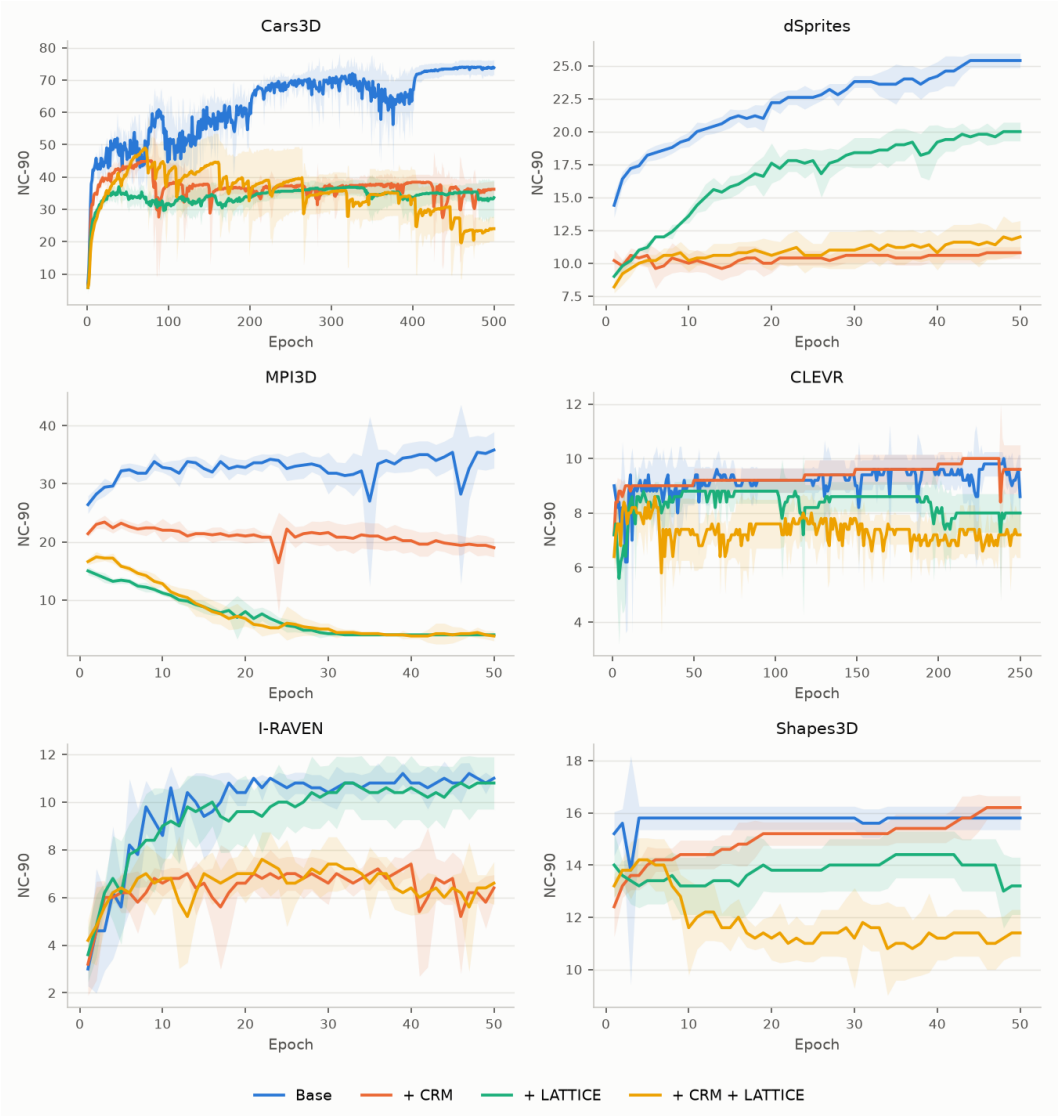

Figure 20: NC-90 over training epochs for ED and versions of ED augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

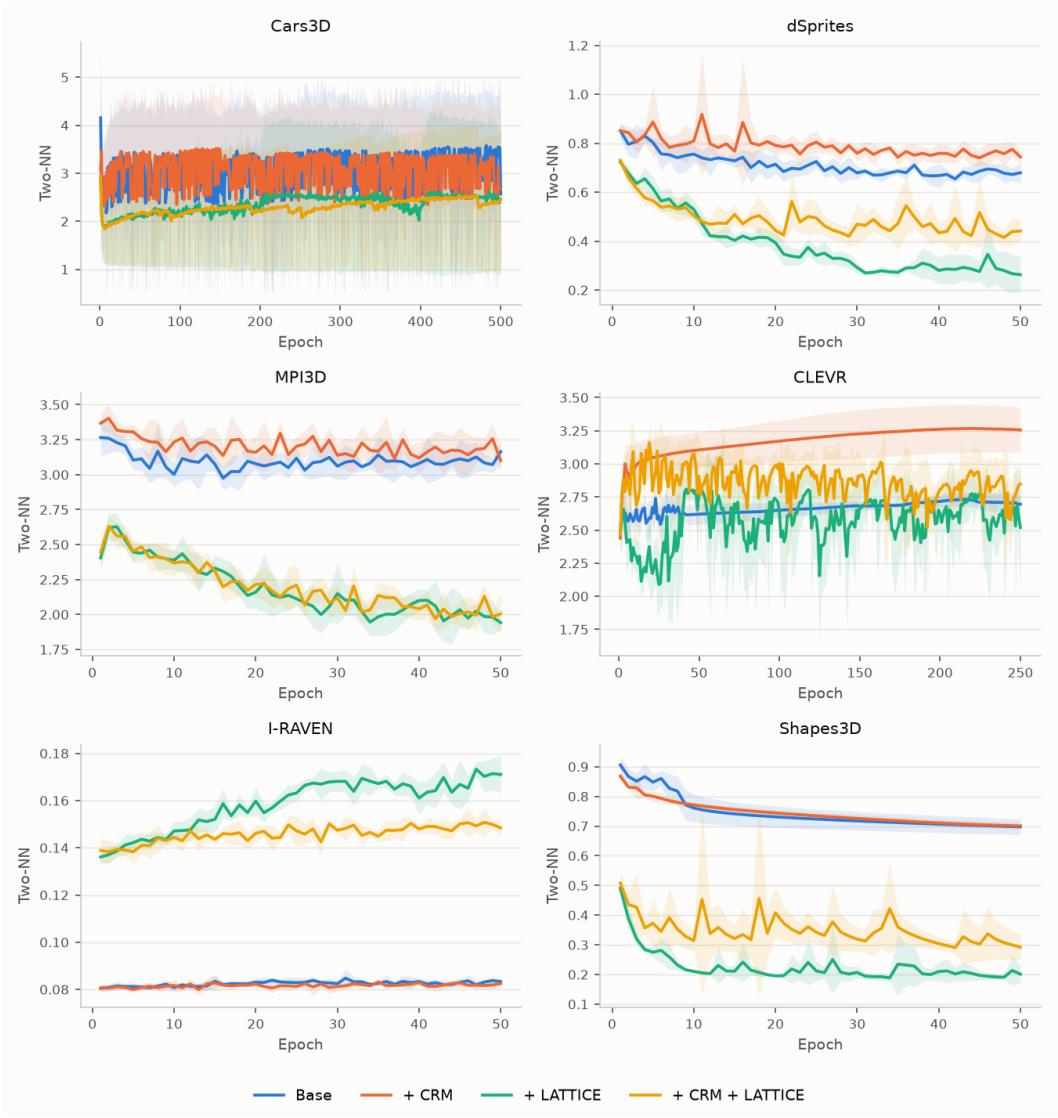

Figure 21: Two-NN over training epochs for ResNet-18 and versions of ResNet-18 augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

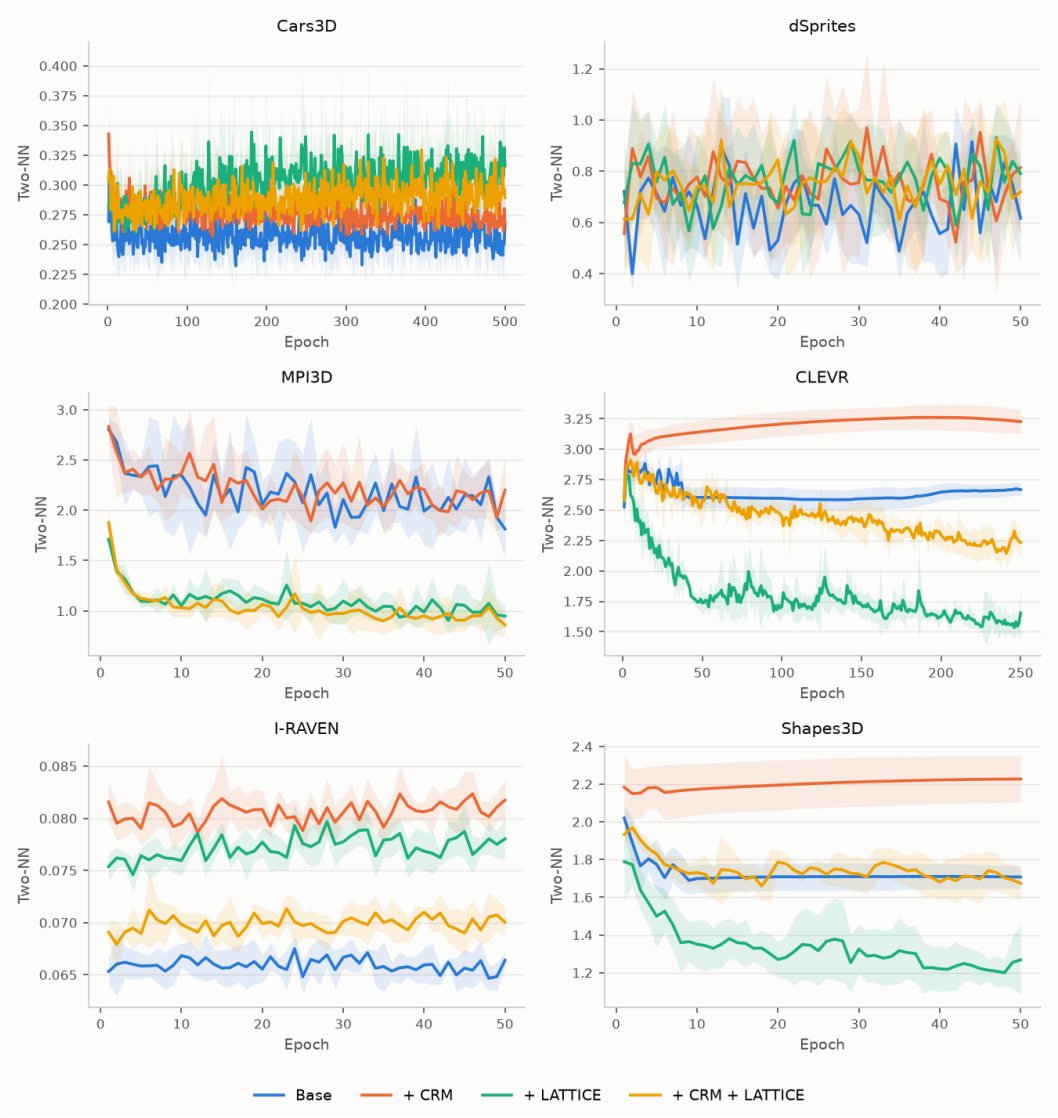

Figure 22: Two-NN over training epochs for ResNet-18 and versions of ResNet-18 augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.



Figure 23: Two-NN over training epochs for AIN and versions of AIN augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

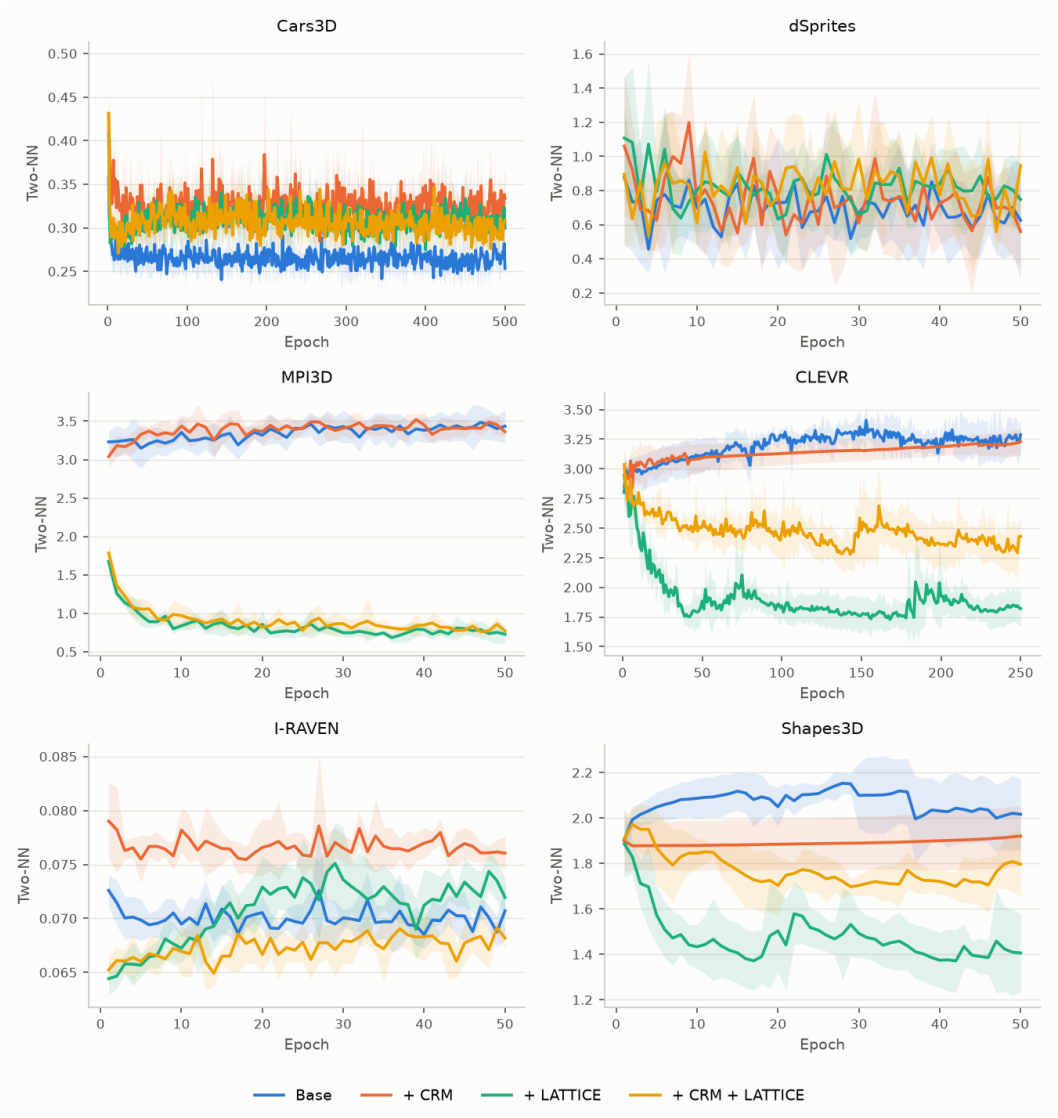

Figure 24: Two-NN over training epochs for AIN and versions of AIN augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

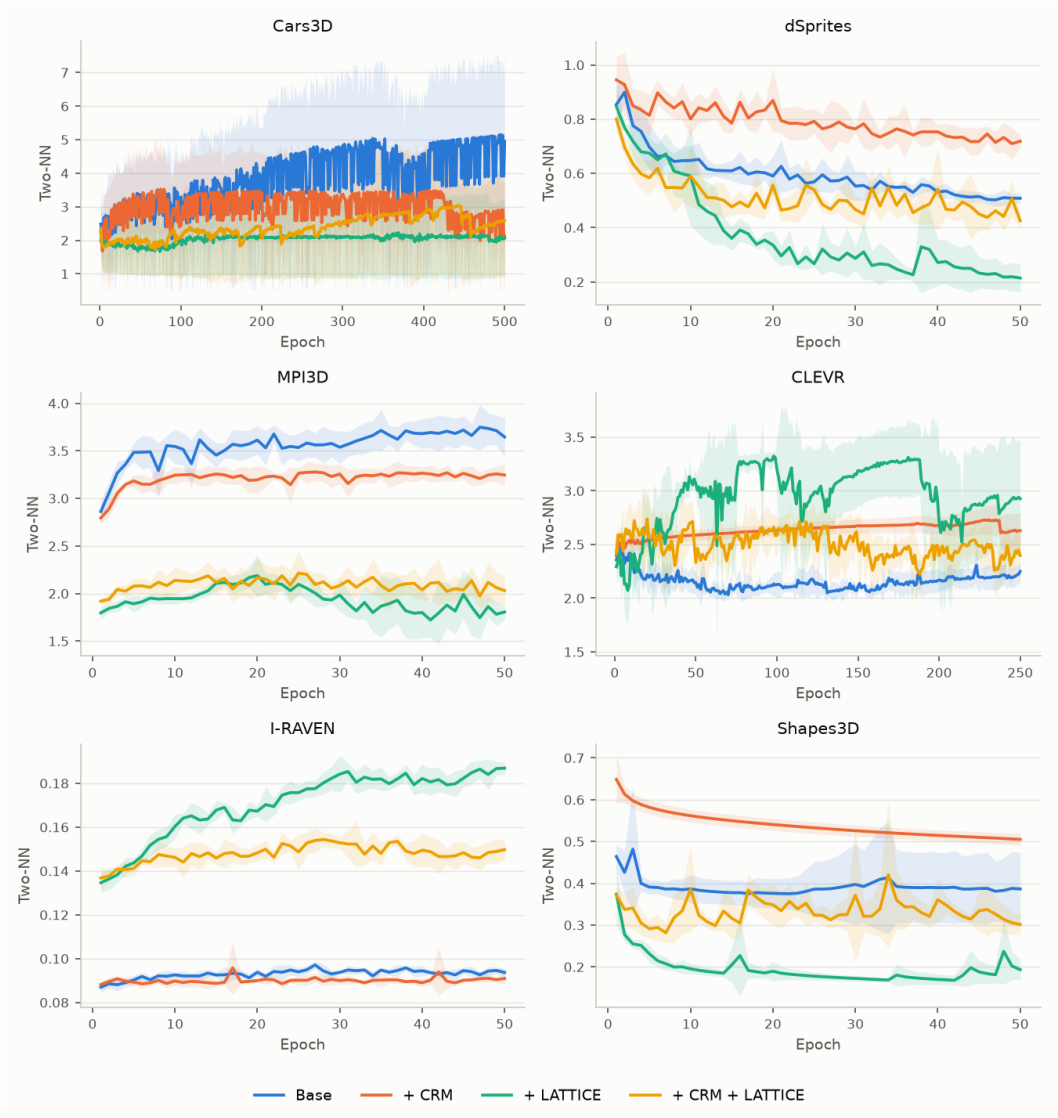

Figure 25: Two-NN over training epochs for ED and versions of ED augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

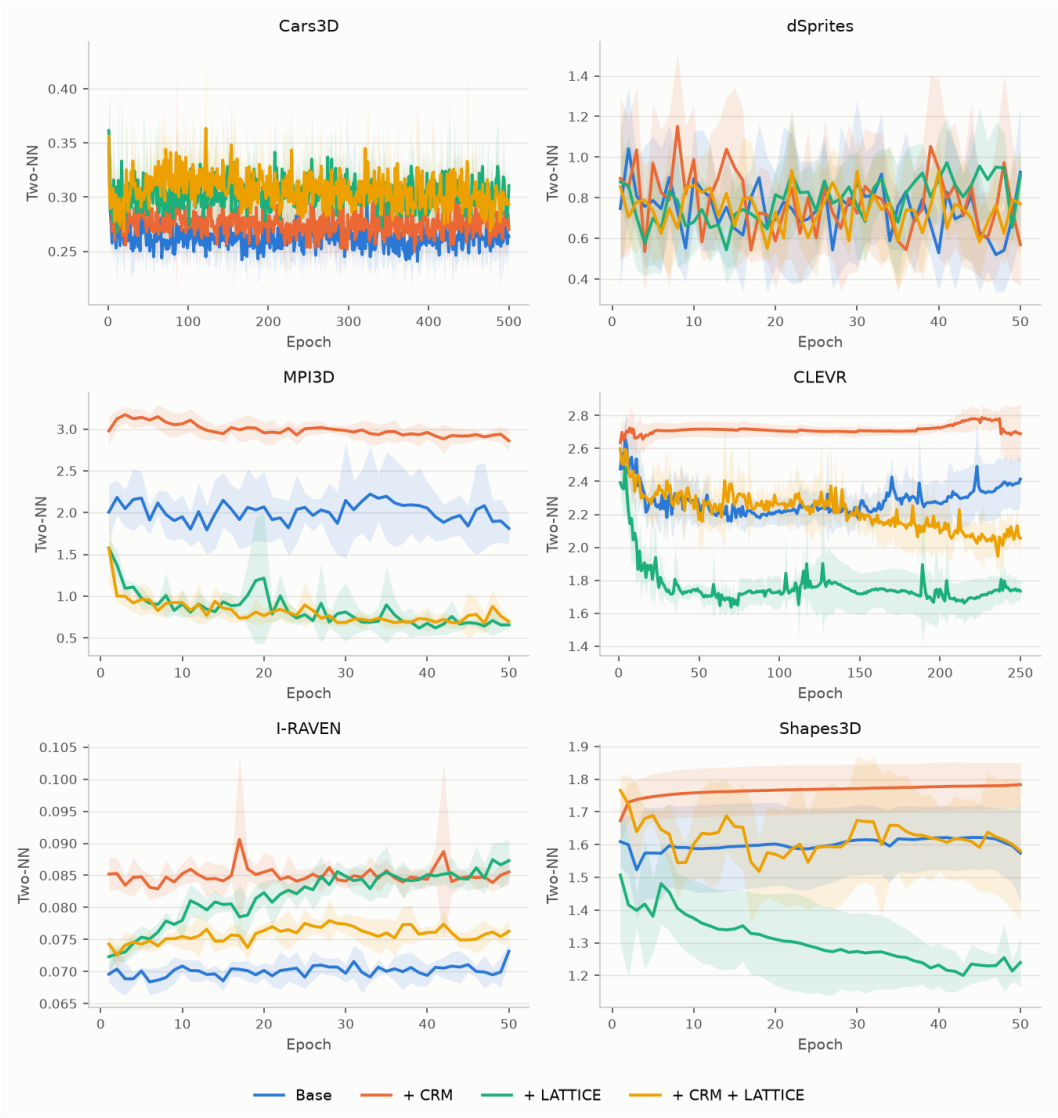

Figure 26: Two-NN over training epochs for ED and versions of ED augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

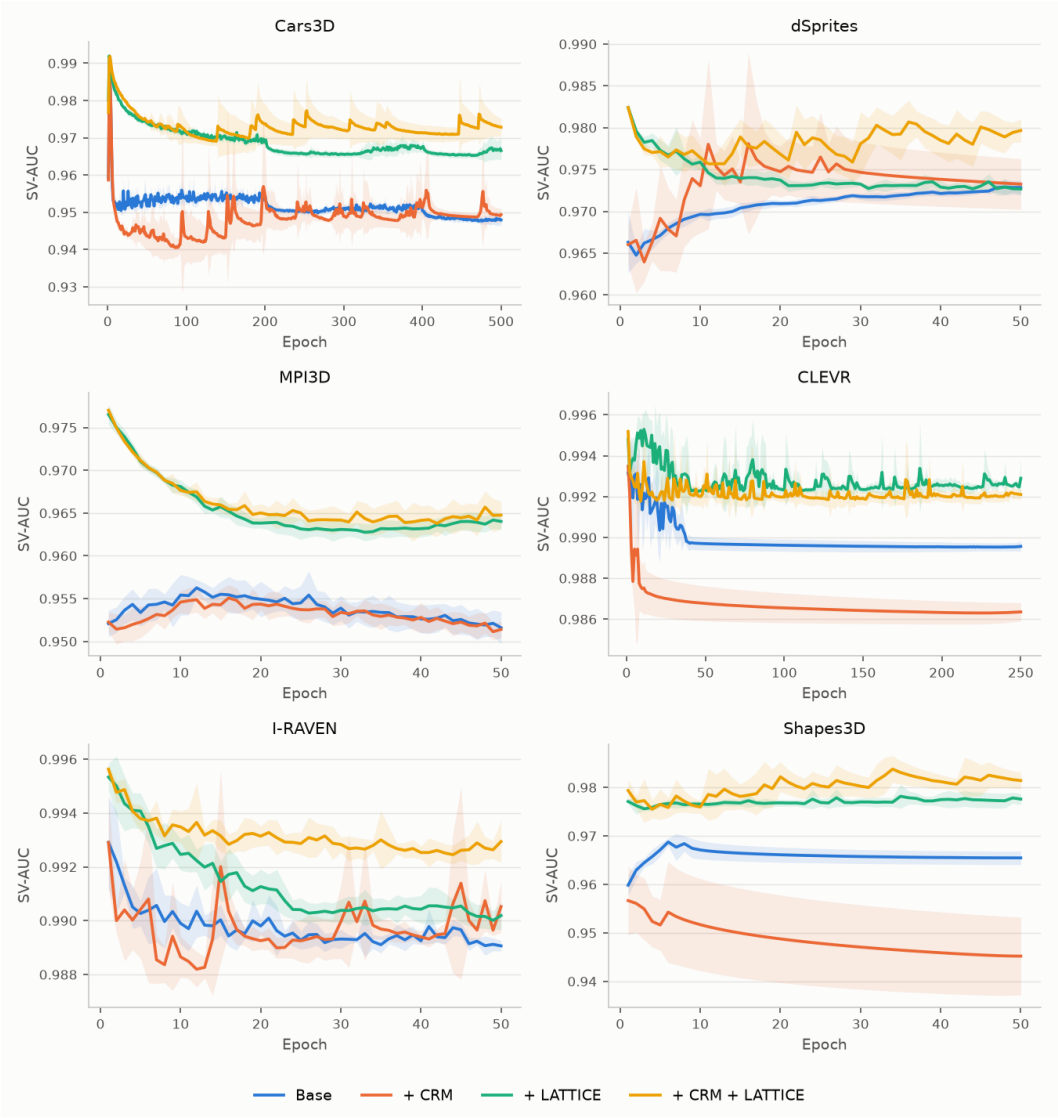

Figure 27: SV-AUC over training epochs for ResNet-18 and versions of ResNet-18 augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.



Figure 28: SV-AUC over training epochs for ResNet-18 and versions of ResNet-18 augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

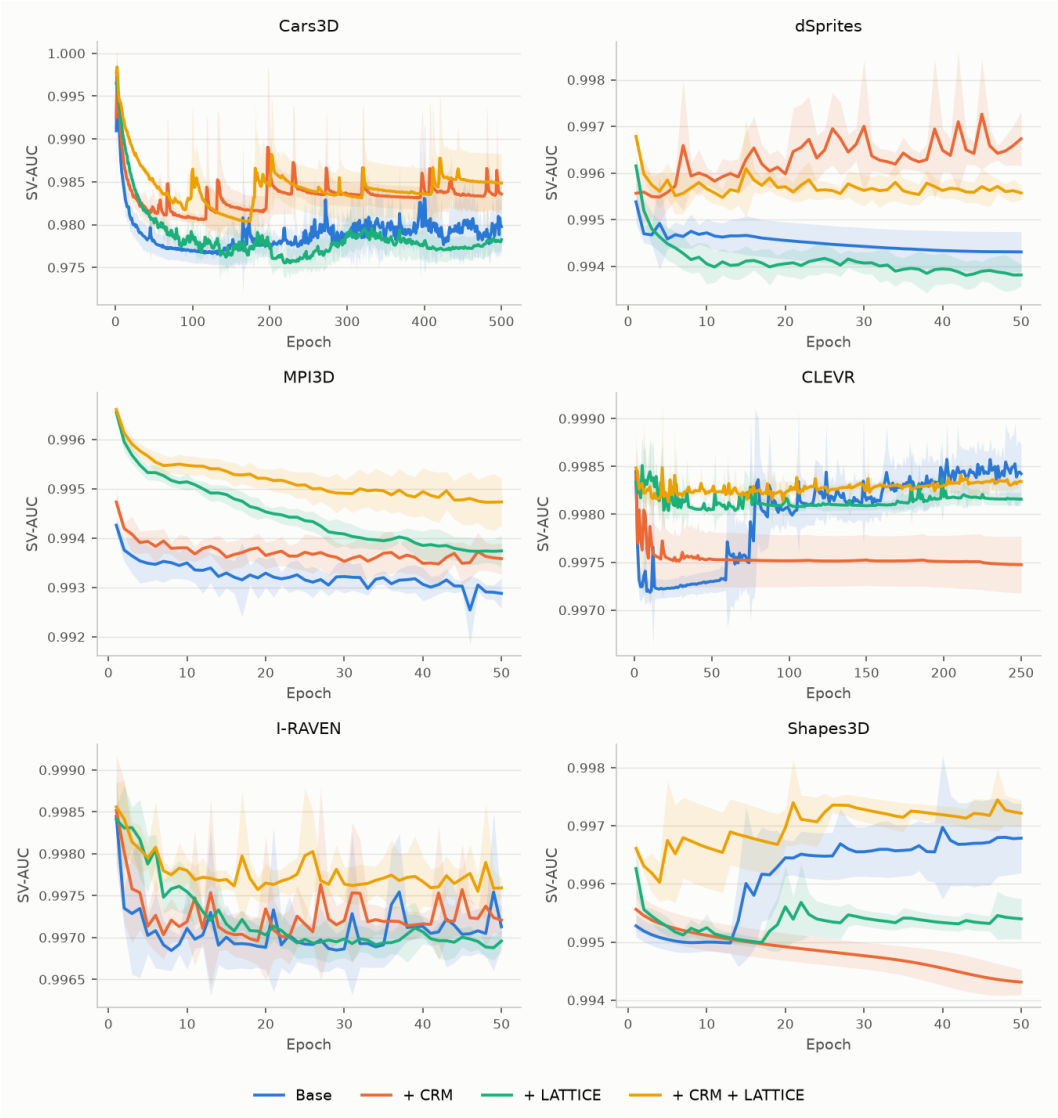

Figure 29: SV-AUC over training epochs for AIN and versions of AIN augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

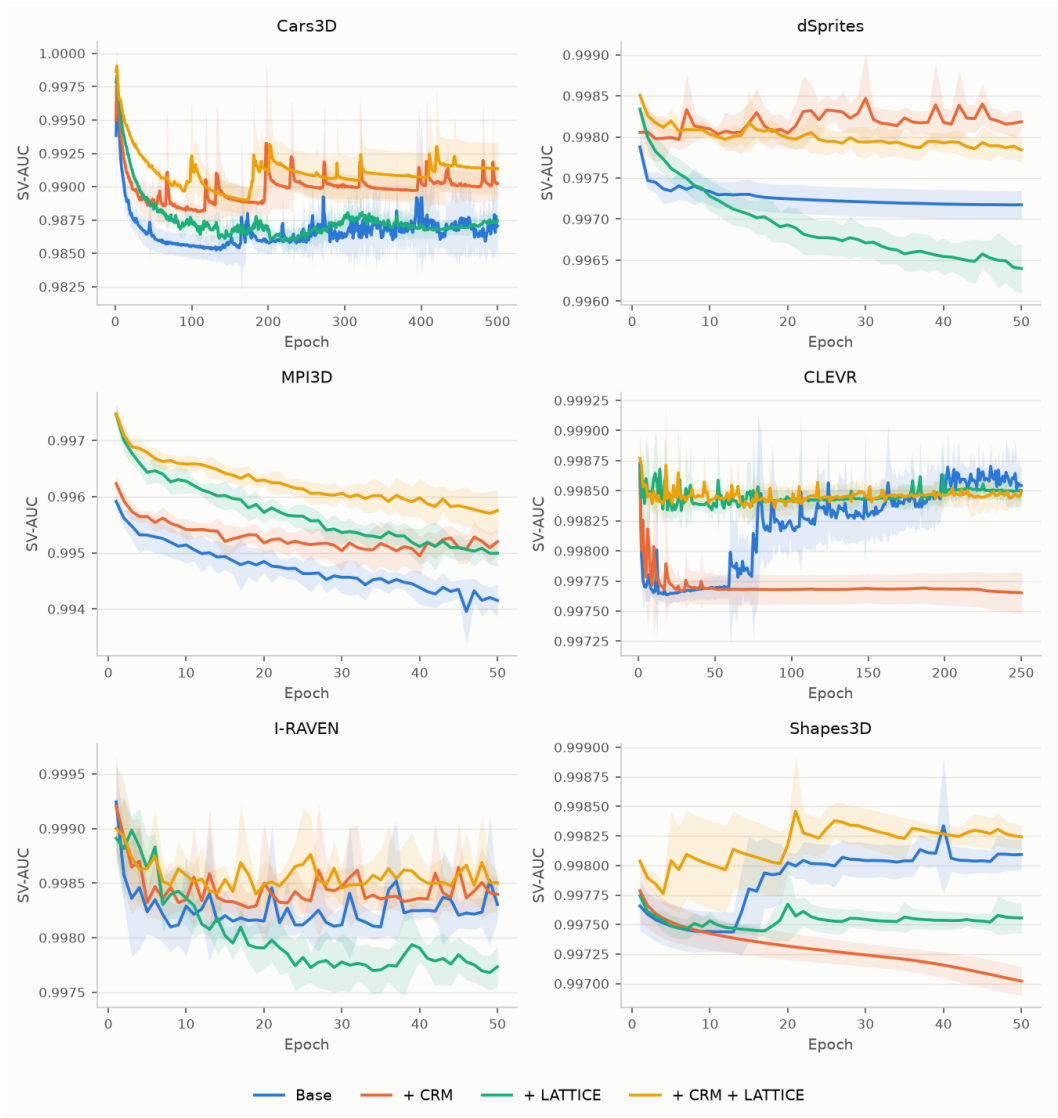

Figure 30: SV-AUC over training epochs for AIN and versions of AIN augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

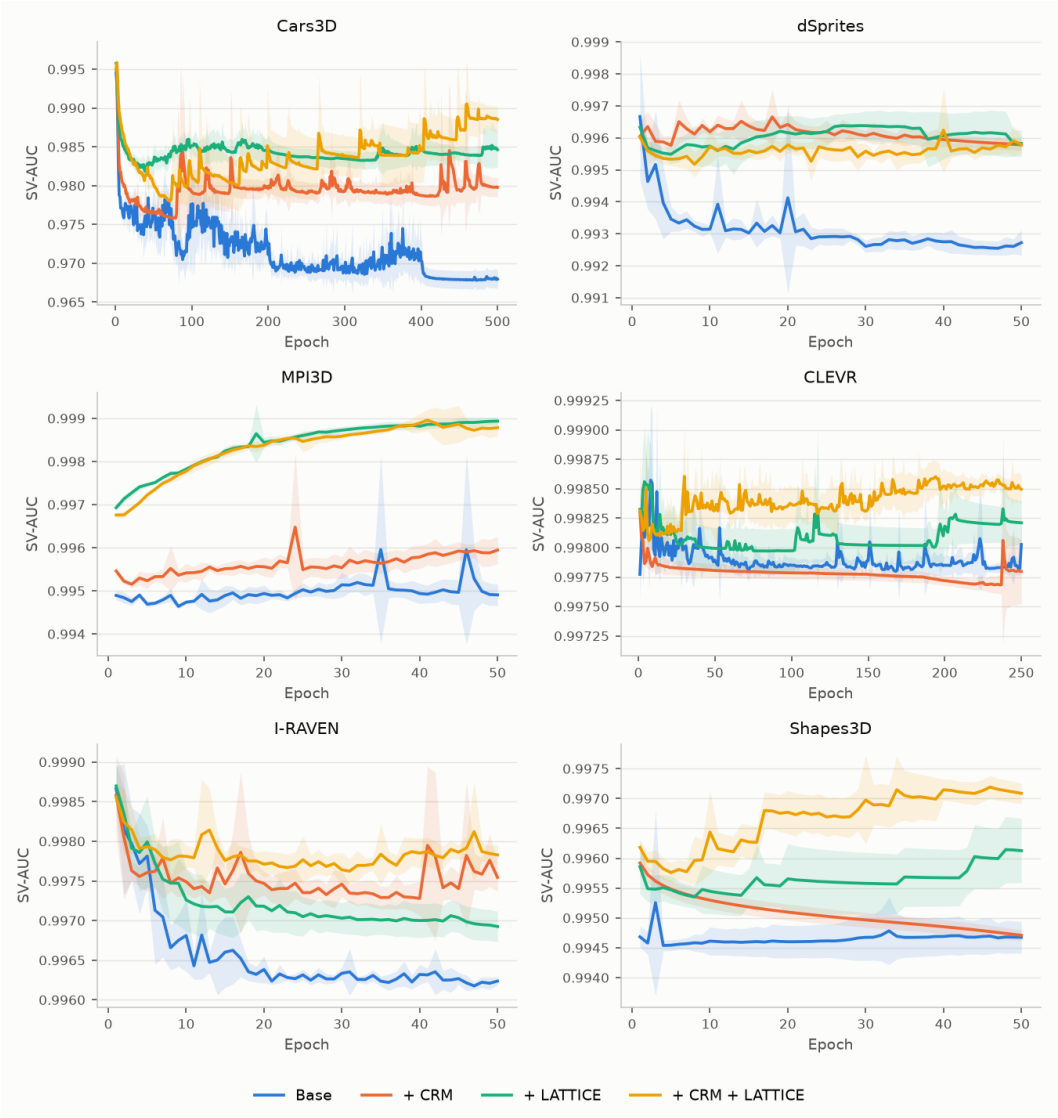

Figure 31: SV-AUC over training epochs for ED and versions of ED augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

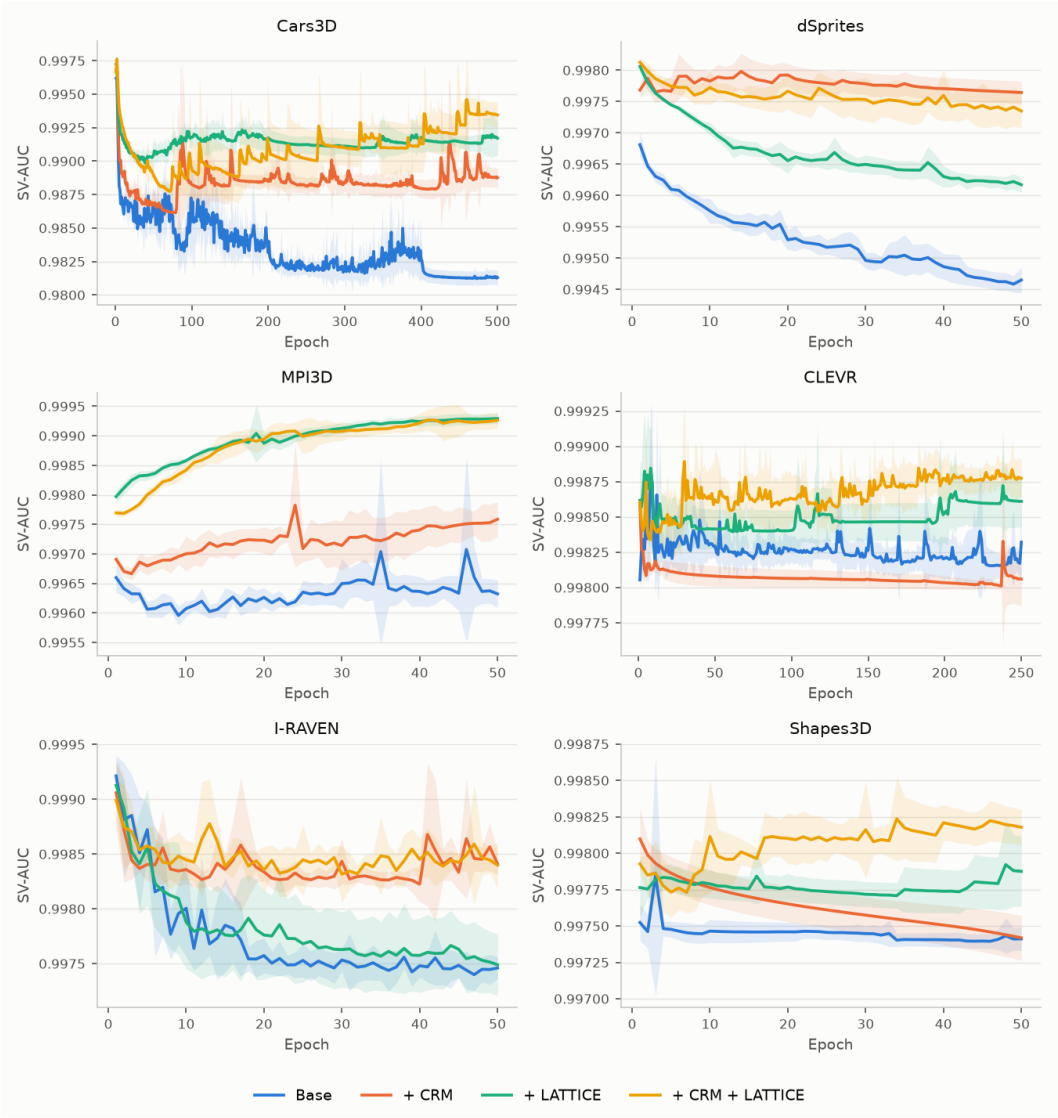

Figure 32: SV-AUC over training epochs for ED and versions of ED augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

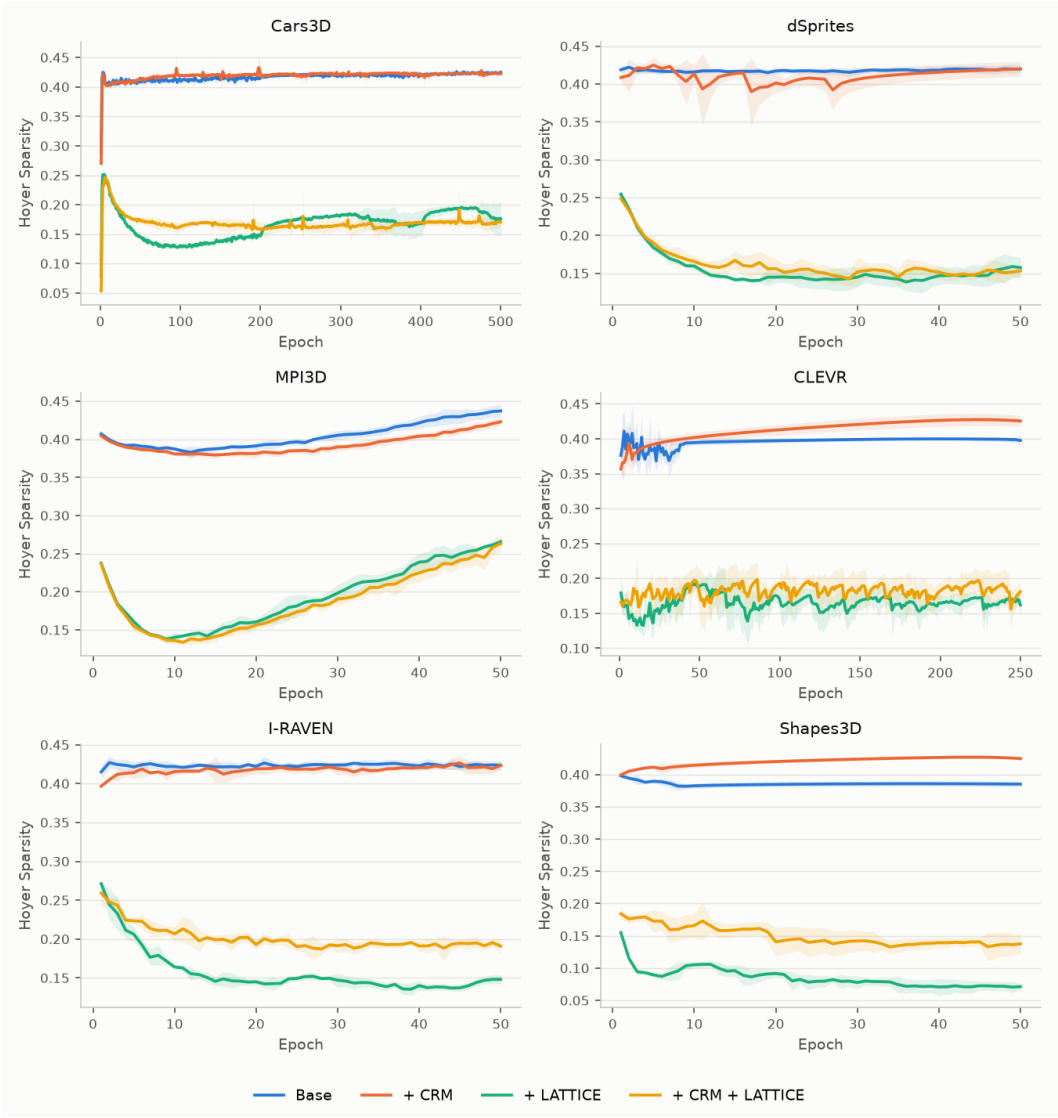

Figure 33: Hoyer Sparsity over training epochs for ResNet-18 and versions of ResNet-18 augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

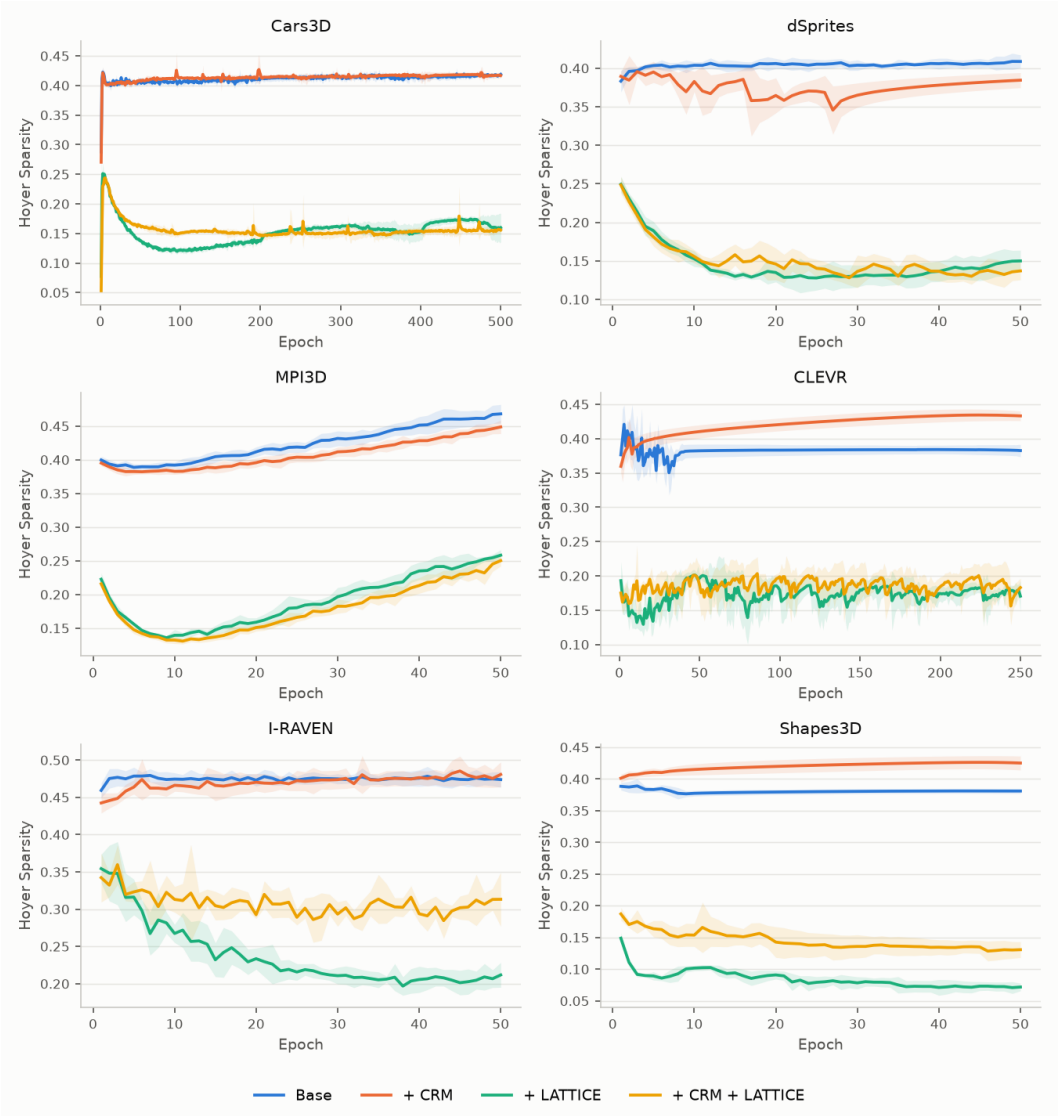

Figure 34: Hoyer Sparsity over training epochs for ResNet-18 and versions of ResNet-18 augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

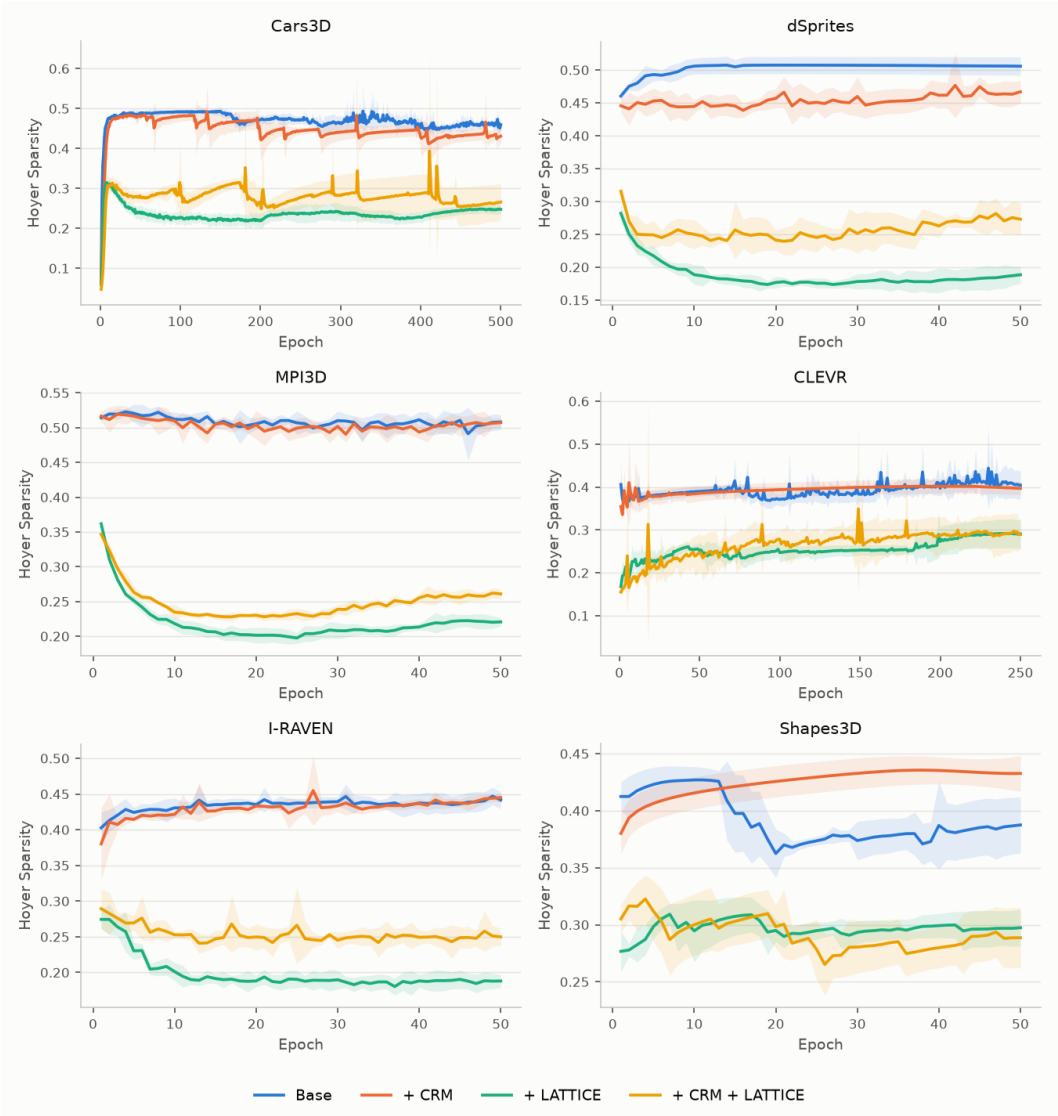

Figure 35: Hoyer Sparsity over training epochs for AIN and versions of AIN augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

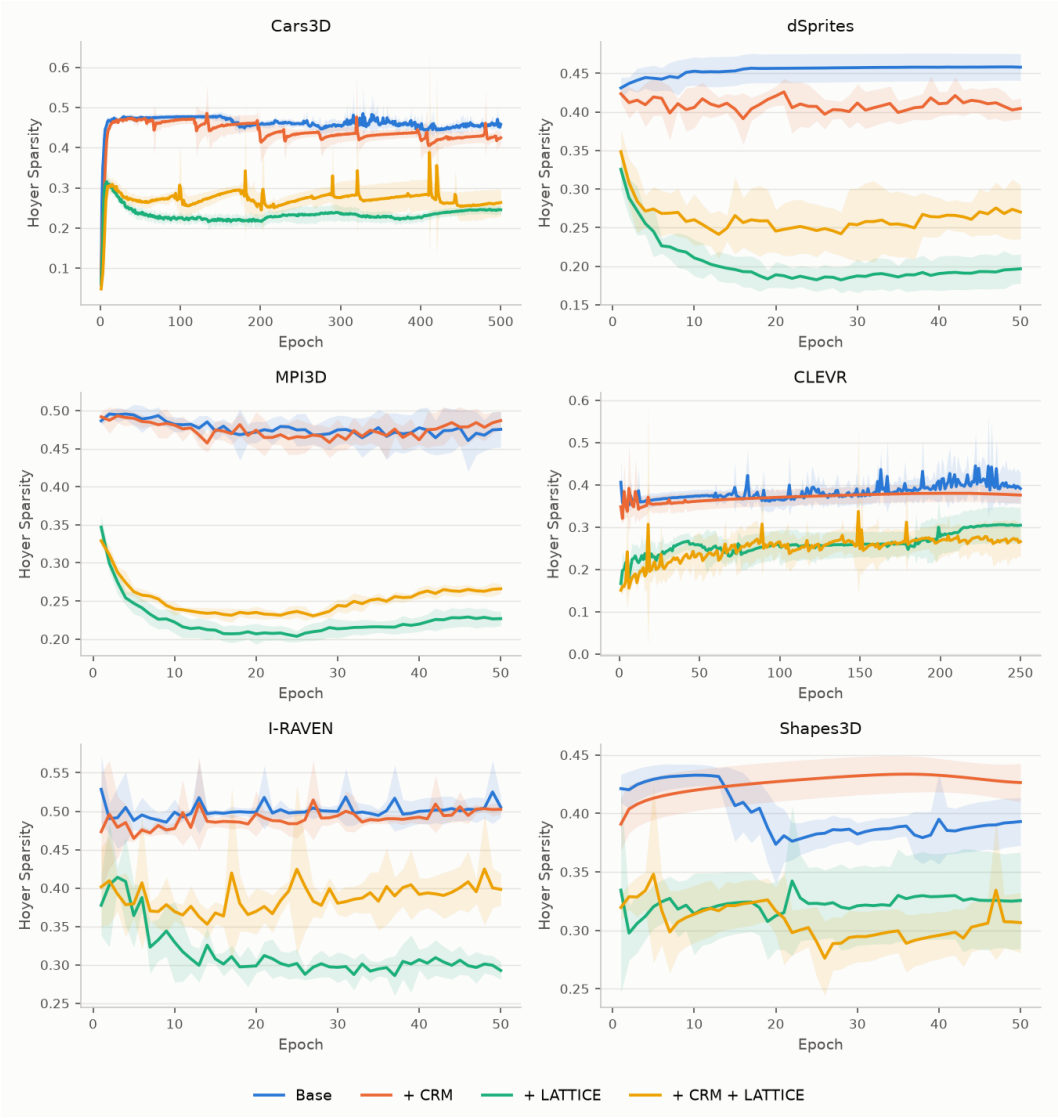

Figure 36: Hoyer Sparsity over training epochs for AIN and versions of AIN augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

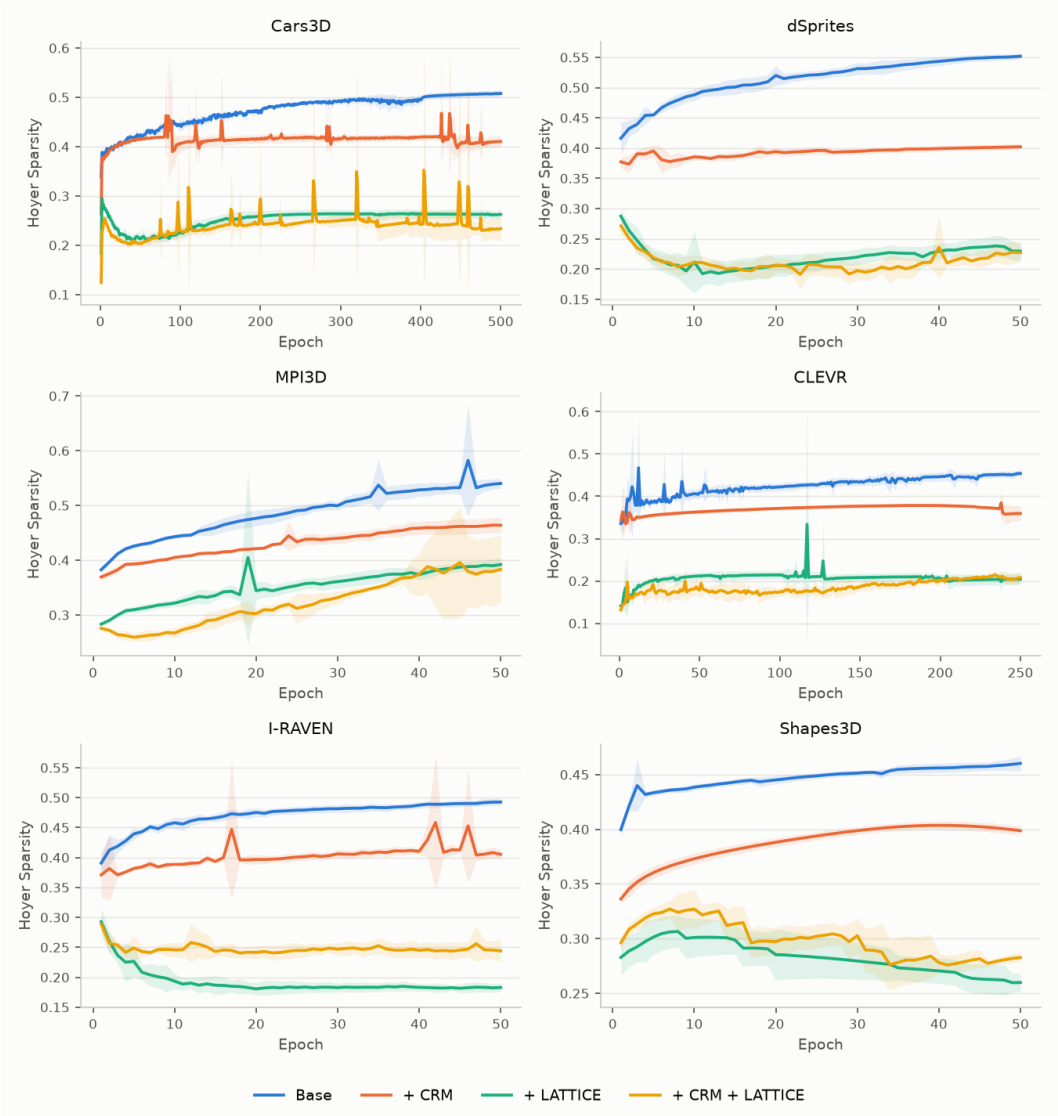

Figure 37: Hoyer Sparsity over training epochs for ED and versions of ED augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

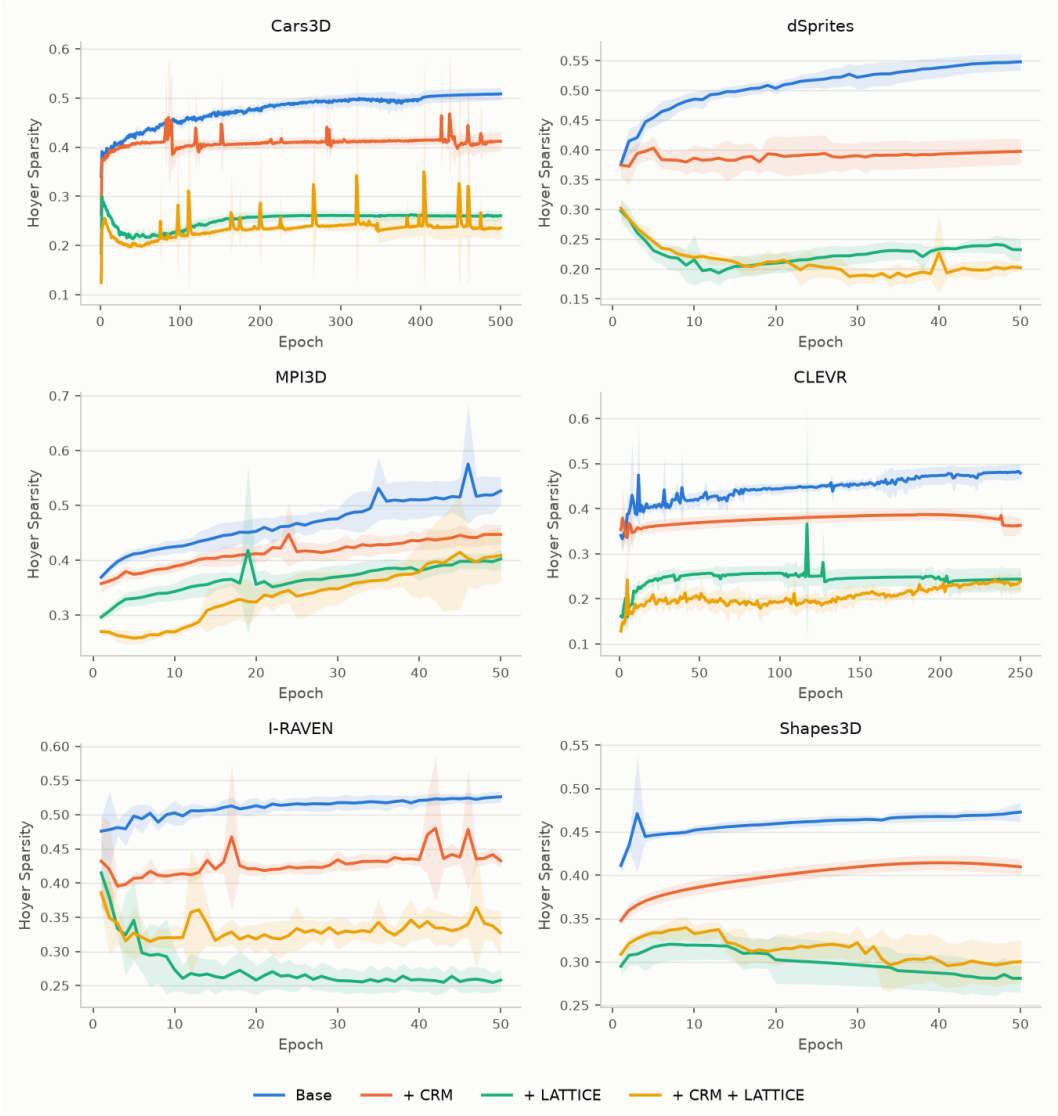

Figure 38: Hoyer Sparsity over training epochs for ED and versions of ED augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

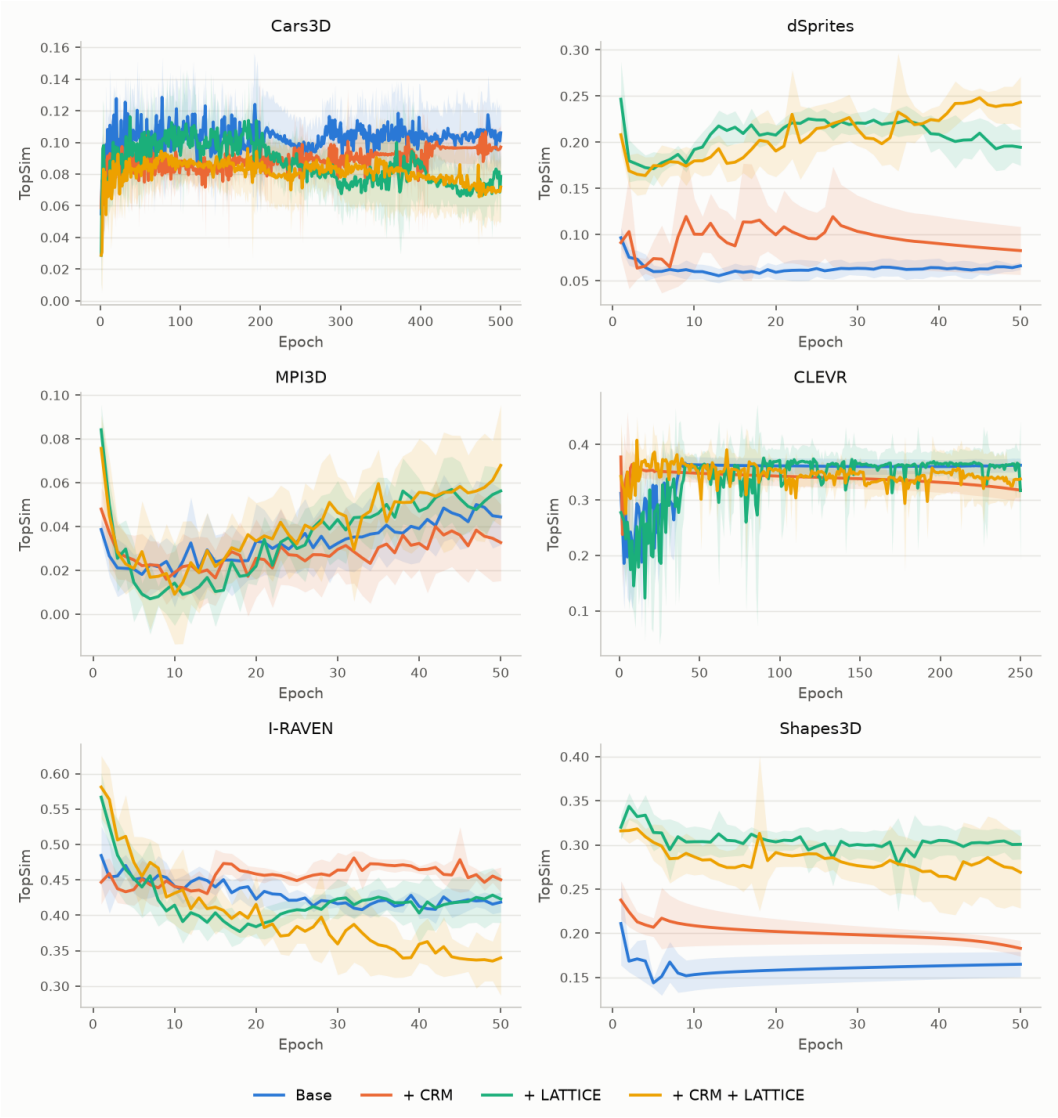

Figure 39: TopSim over training epochs for ResNet-18 and versions of ResNet-18 augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.

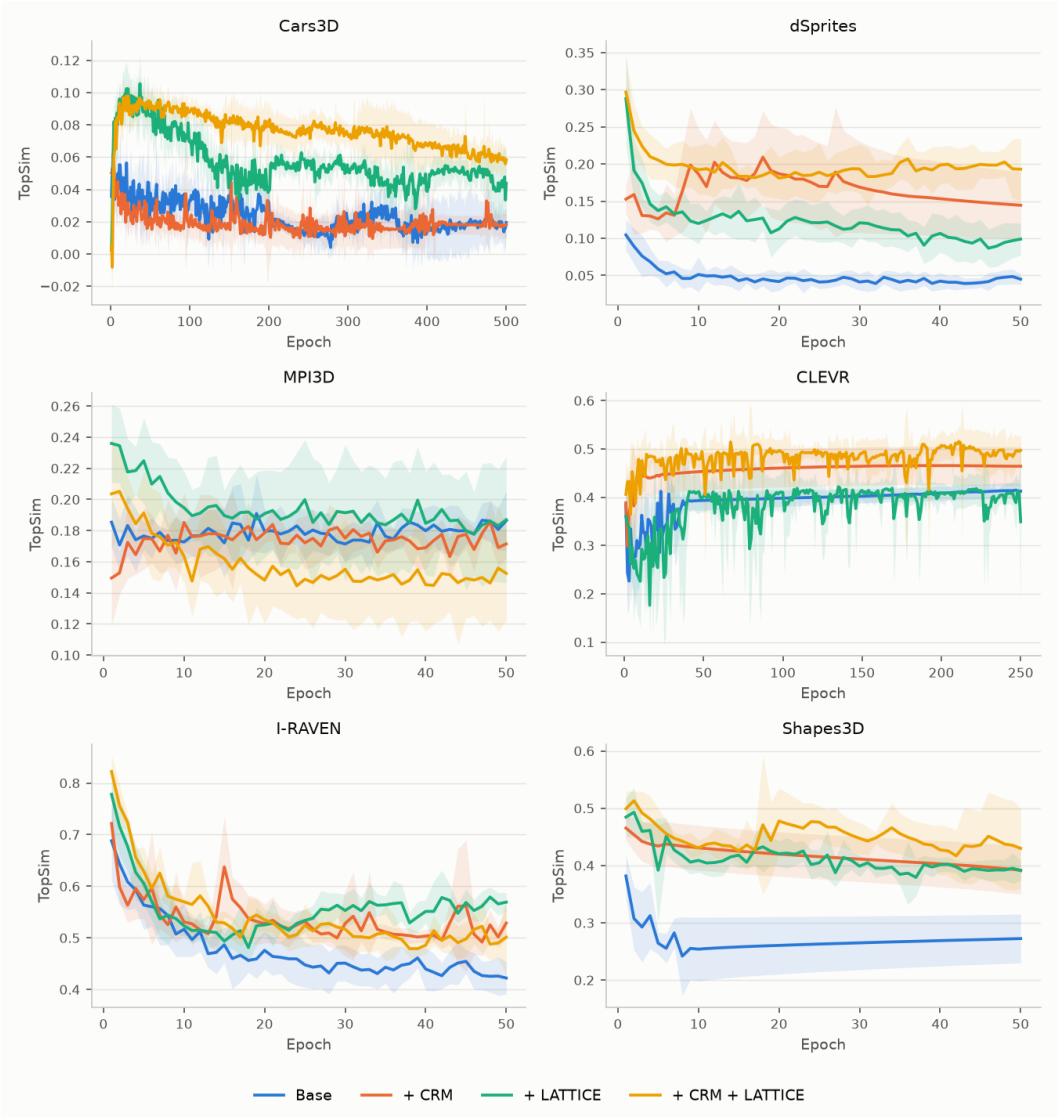

Figure 40: TopSim over training epochs for ResNet-18 and versions of ResNet-18 augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.



Figure 41: TopSim over training epochs for AIN and versions of AIN augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.



Figure 42: TopSim over training epochs for AIN and versions of AIN augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.

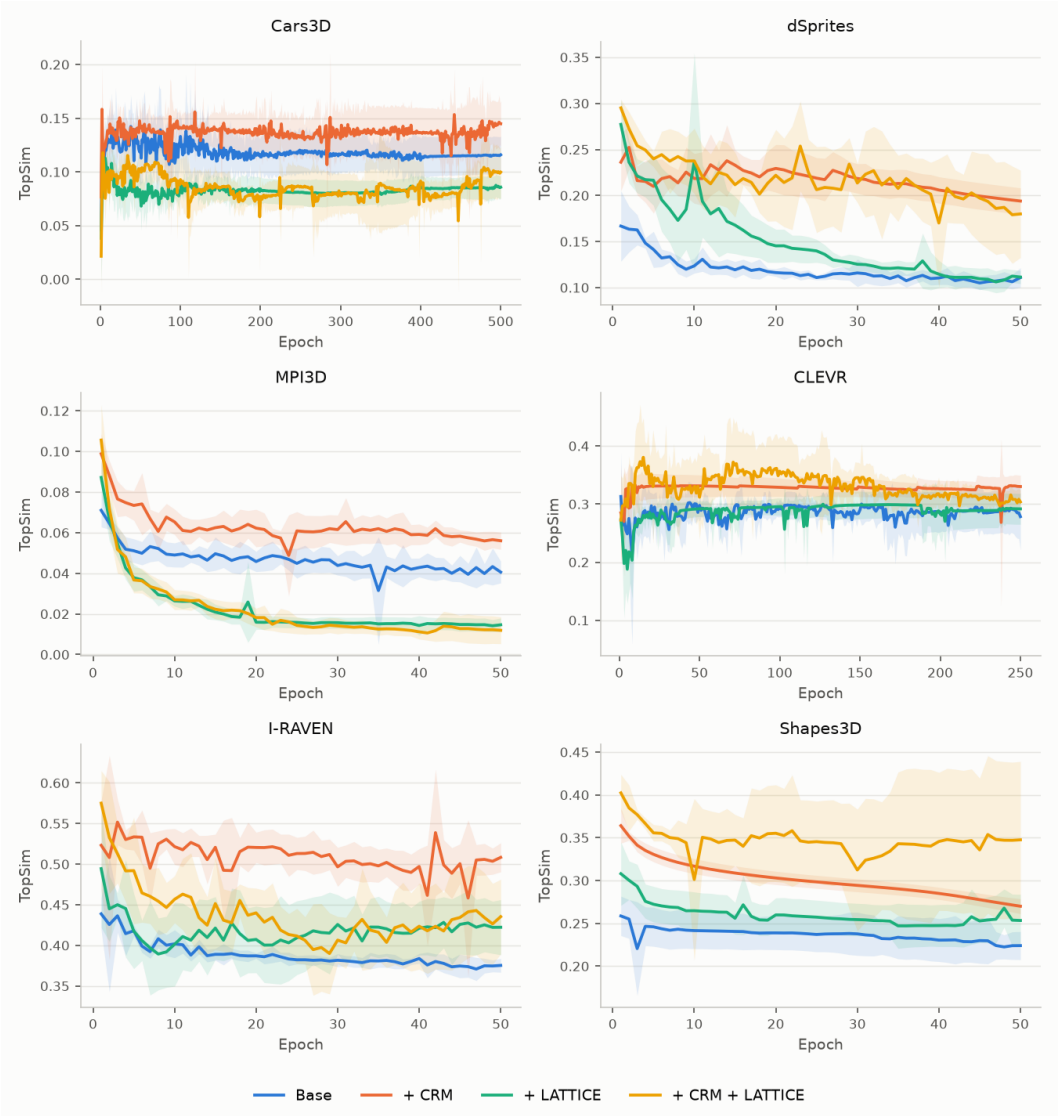

Figure 43: TopSim over training epochs for ED and versions of ED augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the validation set.



Figure 44: TopSim over training epochs for ED and versions of ED augmented with LATTICE and/or CRM. Each subplot corresponds to a different dataset, and each curve corresponds to a model in that family. Values come from the test set.