Identifiable Object-Centric Representation Learning via Probabilistic Slot Attention

Avinash Kori[1*] **Francesco Locatello**[2] **Ainkaran Santhirasekaram**[1]

Francesca Toni[1] **Ben Glocker**[1] **Fabio De Sousa Ribeiro**[1*]

[1] Imperial College London, UK
[2] Institute of Science and Technology, Austria
a.kori21@imperial.ac.uk

Abstract

Learning modular object-centric representations is crucial for systematic generalization. Existing methods show promising object-binding capabilities empirically, but theoretical identifiability guarantees remain relatively underdeveloped. Understanding when object-centric representations can theoretically be identified is crucial for scaling slot-based methods to high-dimensional images with correctness guarantees. To that end, we propose a probabilistic slot-attention algorithm that imposes an *aggregate* mixture prior over object-centric slot representations, thereby providing slot identifiability guarantees without supervision, up to an equivalence relation. We provide empirical verification of our theoretical identifiability result using both simple 2-dimensional data and high-resolution imaging datasets.

1 Introduction

It has been hypothesized that developing machine learning (ML) systems capable of human-level understanding requires imbuing them with notions of *objectness* [48, 65]. Objectness notions can be characterised as physical, abstract, semantic, geometric, or via spaces and boundaries [83, 17]. Humans can generalise across environments with few examples to learn from [70], and this has been attributed to our ability to segregate percepts into object entities [64, 25, 45, 4].

Obtaining object-centric representations is deemed to be a key step for achieving true compositional generalization [5, 48, 3, 22], and uncovering causal influence between discrete concepts and their environment [57, 19, 20, 65, 4]. Significant progress in learning object-centric representations has been made [15, 16, 44, 66, 10, 67], particularly in unsupervised object discovery settings using an iterative attention mechanism known as Slot Attention (SA) [53]. However, most existing work approaches object-centric representation learning empirically, leaving theoretical understanding relatively underdeveloped. Establishing the *identifiability* [31, 29] of representations is important as it clarifies under which conditions object-centric representation learning is theoretically possible [6].

A well-known result shows that identifiability of latent variables is fundamentally impossible without assumptions about the data generating process [31, 51]. *Therefore, understanding when object representations can theoretically be identified is important to scale object-centric methods to high-dimensional images.* Recent works [6, 47] make important advances on this by explicitly

[*] Equal Contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

stating the set of assumptions necessary for providing theoretical identifiability of object-centric representations. However, they restrict their attention to properties of the *mixing function*, studying a class of models with *additive* decoders. Although there are merits to this approach, there are practical challenges with the so-called *compositional contrast* objective [6], as it involves computing Jacobians and requires second-order optimization via gradient descent. Consequently, satisfying the identifiability conditions explicitly (e.g. compositional contrast must be zero) is computationally restrictive for moderately high-dimensional data. In this work, we present a probabilistic perspective that is not subject to the same scalability issues while still providing theoretical identifiability of object-centric representations without supervision. In Table 1, we list object-centric learning methods, their (sometimes implicit) modelling assumptions (see § 5 for additional information and Appendix A for a detailed breakdown and discussion), and their respective identifiability guarantees of object representations. Most methods do not guarantee identifiability, and make the $\mathcal{B}$-disentanglement (1) and *additive decoder* (2) assumptions. Brady *et. al.* [6] do provide identifiability guarantees and additionally assume *irreducibility* (3) and *compositionality* (4). Our method provides identifiability guarantees by introducing latent structure (i.e. via a GMM prior) which generalizes to non-additive decoders. This is advantageous as the computational complexity of additive decoders scales linearly with the number of slots K – whereas our approach is invariant to K. Moreover, non-additive decoders have been found to significantly improve performance in practice [66, 67, 69], though the theoretical basis is underexplored. Finally, latent structure can reduce the complexity burden on the mixing function f (decoder), making it easier to learn in practice [18, 43].

Table 1: Identifiability strategies (mixing function f or latent dist. $p(\mathbf{z})$), and assumptions made by object-centric learning methods.

METHOD	ASSUMPTION	IDENTIF.
[9, 14, 15, 21, 53, 76]	1, 2	N/A
[13, 42, 50, 73]	1, 2, 5	N/A
[44]	1, 2, 6	N/A
[6]	1, 2, 3, 4, 7	f
Proposed	1, 8	$p(\mathbf{z})$

Contributions. Our main contributions are the following: **(i)** We prove that object-centric representations (i.e. slots) are identifiable without supervision up to an equivalence relation (§ 5) under a latent mixture model specification. To that end, we propose a probabilistic slot-attention algorithm (§ 4) which imposes an *aggregate* mixture prior over slot representations. **(ii)** We show that our approach induces a non-degenerate (*global*) Gaussian Mixture Model (GMM) by aggregating per-datapoint (*local*) GMMs, providing a slot prior which: (a) is empirically stable across runs (i.e. identifiable up to affine transformations and slot permutations); (b) can be tractably sampled from. **(iii)** We provide conclusive empirical evidence of our theoretical object-centric identifiability result, including visual verification on synthetic 2-dimensional data as well as standard imaging benchmarks (§ 6).

2 Related Work

Identifiable Representation Learning. Identifiability of representations stems from early work in independent component analysis (ICA) [31, 29], and is making a resurgence recently [30, 32, 51, 37, 75, 46, 82]. Common strategies for tackling this identifiability problem are: (i) restricting the class of mixing functions; (ii) using non-i.i.d data, interventional data or counterfactuals; and (iii) imposing structure in the latent space via distributional assumptions. Regarding (i), restricting the class of the mixing functions to conformal maps [8] or volume-preserving transformations [81] has been found to produce identifiable models. For (ii), prior works [84, 52, 7, 2, 75] assume access to contrastive pairs of observations $(\mathbf{x}, \tilde{\mathbf{x}})$ obtained from either data augmentation, interventions, or approximate counterfactual inference. As for (iii), latent space structure is enforced via either: (a) using auxiliary variables to make latent variables conditionally independent [33, 37, 38]; or (b) distributional assumptions such as placing a mixture prior over the latent variables in a VAE [11, 80, 43]. In this work, we prove an identifiability result via strategy (iii) but within an object-centric learning context, where the latent variables are a set of object *slots* [53].

Object-Centric Learning. Much early work on unsupervised representation learning is based on the Variational Autoencoder (VAE) framework [41], and relies on independence assumptions between latent variables to learn so-called *disentangled* representations [5, 24, 40, 12, 59]. These methods are closely linked to object-centric representation learning [9, 15, 21], as they also leverage (iterative) variational inference procedures [58, 73, 50]. Alternatively, an iterative attention mechanism known as slot attention (SA) [53] has been the focus of much follow-up work recently [16, 67, 76, 68, 14].

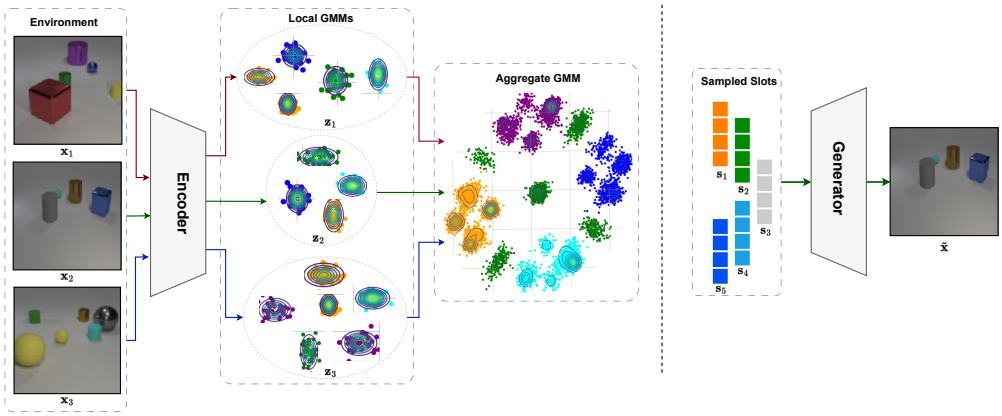


Figure 1: **Probabilistic slot attention and the identifiable aggregate slot posterior**. (Left) Slot posterior GMMs per datapoint (local) and the induced *aggregate* posterior GMM (global). (Right) Sampling slot representations from the aggregate slot posterior is tractable.

Although slot attention-based methods show promising object *binding* [22] capabilities empirically on select datasets, they do not provide identifiability guarantees on the learned representations. Recently, [55, 56] assume the access to interventional data-generating process following [1] to demonstrate the identifiability of object-centric representations, while [6, 47] presented the identifiability results for object representations (i.e. slots), clarifying the necessary assumptions and properties of the *mixing function* (e.g. additive decoders). However, satisfying Brady *et. al.* [6]'s *compositional contrast* identifiability condition explicitly (must be zero) requires computationally restrictive second-order optimization. In contrast, we shift the focus to learning structured object-centric latent spaces via Probabilistic Slot Attention (PSA), bridging the gap between generative model identifiability literature and object-centric representation learning. Notably, our PSA approach is also related to probabilistic capsule routing [27, 63, 62, 61] since slots are equivalent to *universal* capsules [26], but like slot attention, offers output permutation symmetry and does not face scalability issues.

3 Background

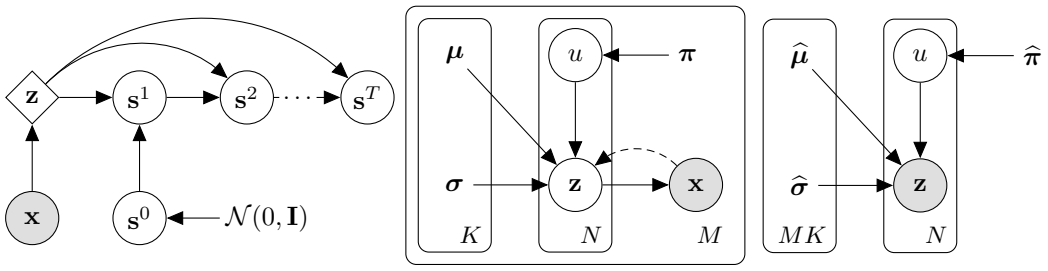
Notation. Let $\mathcal{X} \subseteq \mathbb{R}^{H \times W \times C}$ denote the input image space, where each image $\mathbf{x}$ is of size $H \times W$ pixels with C channels. Let $f_e : \mathcal{X} \to \mathcal{Z}$ denote an encoder mapping image space to a latent space $\mathcal{Z} \subseteq \mathbb{R}^{N \times d}$, where each latent variable $\mathbf{z}$ consists of N, d-dimensional vectors. Lastly, let $f_d : \mathcal{S} \to \mathcal{X}$ denote a decoder mapping from slot representation space $\mathcal{S} \subseteq \mathbb{R}^{K \times d}$ to image space.

Slot Attention. Slot attention [53] receives a set of feature embeddings $\mathbf{z} \in \mathbb{R}^{N \times d}$ per input $\mathbf{x}$, and applies an iterative attention mechanism to produce K object-centric representations called slots $\mathbf{s} \in \mathbb{R}^{K \times d}$. Let $\boldsymbol{W}_k, \boldsymbol{W}_v$ denote *key* and *value* transformation matrices acting on $\mathbf{z}$, and $\boldsymbol{W}_q$ the *query* transformation matrix acting on $\mathbf{s}$. To simplify our exposition later on, let $f_s : \mathcal{Z} \times \mathcal{S} \to \mathcal{S}$ be shorthand notation for the *slot update* function, defined as:

$$\mathbf{s}^{t+1} := f_s(\mathbf{z}, \mathbf{s}^t) = \hat{\boldsymbol{A}}\mathbf{v}, \qquad \hat{A}_{ij} := \frac{A_{ij}}{\sum_{l=1}^{N} A_{il}}, \qquad \boldsymbol{A} := \text{softmax}\left(\frac{\mathbf{q}\mathbf{k}^T}{\sqrt{d}}\right) \in \mathbb{R}^{K \times N}, \tag{1}$$

where $\mathbf{q} = \boldsymbol{W}_q \mathbf{s}^t \in \mathbb{R}^{K \times d}$, $\mathbf{k} = \boldsymbol{W}_k \mathbf{z} \in \mathbb{R}^{N \times d}$, and $\mathbf{v} = \boldsymbol{W}_v \mathbf{z} \in \mathbb{R}^{N \times d}$ correspond to the query, key and value vectors respectively and $\boldsymbol{A} \in \mathbb{R}^{K \times N}$ is the attention matrix. Unlike self-attention [74], the queries $\mathbf{q}$ in slot attention are a function of the slots $\mathbf{s}^t$, and are iteratively refined over T iterations. The initial slots $\mathbf{s}^{t=0}$ are randomly sampled from a standard Gaussian. The queries at iteration t are given by $\hat{\mathbf{q}}^t = \boldsymbol{W}_q \mathbf{s}^t$, and the slot update process can be summarized as in Equation 1.

Compositionality. Compositionality as defined by Brady et al. [6] is a structure imposed on the slot decoder mapping f_d which implies that each image pixel is a function of at most one slot representation, thereby enforcing a local sparsity structure on the Jacobian matrix of f_d.



(a) Slot Attention Encoder (b) Probabilistic Slot Attention (c) Aggregate Posterior Mixture

Figure 2: **Graphical models of probabilistic slot attention**. **(a)** Stochastic encoder of standard slot attention [53] with T attention iterations. **(b)** Proposed model – each image in the dataset $\{\mathbf{x}_i\}_{i=1}^M$ is encoded into a respective latent representation $\mathbf{z} \in \mathbb{R}^{N \times d}$, to which a (local) Gaussian mixture model with K components is fit via expectation maximisation. The resulting K Gaussians serve as slot posterior distributions: $\mathbf{s}_k \sim \mathcal{N}(\mathbf{s}_k; \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2)$, for $k = 1, \ldots, K$. **(c)** Aggregate posterior distribution obtained by marginalizing out the data: $q(\mathbf{z}) = \sum_{i=1}^M q(\mathbf{z} \mid \mathbf{x}_i)/M$. We prove that $q(\mathbf{z})$ is a tractable, non-degenerate Gaussian mixture distribution which: (i) serves as the theoretically optimal prior over slots; (ii) is empirically stable across runs (i.e. identifiable up to an affine transformation and slot permutation); (iii) can be tractably sampled from and used for scene composition tasks.

Definition 1 (Compositional Contrast)**.** For a differentiable mapping $f_d : \mathcal{Z} \to \mathcal{X}$, the compositional contrast of f_d at $\mathbf{z}$ is given by:

$$C_{\text{comp}}(f_d, \mathbf{z}) = \sum_{n=0}^{N} \sum_{k=1}^{K} \sum_{j=k+1}^{K} \left\| \frac{\partial f_d(\mathbf{z})_n}{\partial \mathbf{z}_k} \right\| \left\| \frac{\partial f_d(\mathbf{z})_n}{\partial \mathbf{z}_j} \right\|.$$

Brady et al. [6]'s main result (Theorem 1) relies on *compositionality* and *invertibility* of f_d to guarantee slot-identifiability when both the compositional contrast and the reconstruction loss equal zero. However, using $C_{\text{comp}}(f_d, \mathbf{z})$ as a metric or as part of an objective function is computationally prohibitive[2]. Our method aims to minimize $C_{\text{comp}}(f_d, \mathbf{z})$ implicitly [47], without additive decoders.

4 Probabilistic Slot Attention

In this section, we present a probabilistic slot attention framework which imposes a mixture prior structure over the slot latent space. This structure will prove to be instrumental in establishing our main identifiability result in Section 5. We begin by approaching standard slot attention [53] from a graphical modelling perspective. As shown in Figure 2 and explained in Section 3, applying slot attention to a deterministic encoding $\mathbf{z} = f_e(\mathbf{x}) \in \mathbb{R}^{N \times d}$ yields a set of K object slot representations $\mathbf{s}_{1:K} \coloneqq \mathbf{s}_1, \ldots, \mathbf{s}_K$. This process induces a stochastic encoder $q(\mathbf{s}_{1:K} \mid \mathbf{x})$, where the stochasticity comes from the random initialization of the slots: $\mathbf{s}_{1:K}^{t=0} \sim \mathcal{N}(\mathbf{s}_{1:K}; 0, \mathbf{I}) \in \mathbb{R}^{K \times d}$. Since each slot is a deterministic function of its previous state $\mathbf{s}^t \coloneqq f_s(\mathbf{z}, \mathbf{s}^{t-1})$ it is possible to randomly sample initial states $\mathbf{s}^0$ and obtain stochastic estimates of the slots.[3] However, since each transition depends on $\mathbf{z}$, which in turn depends on the input $\mathbf{x}$, *we do not get a generative model we can tractably sample from*. This can conceivably be remedied by placing a tractable prior over $\mathbf{z}$ and using the VAE framework along the lines of [77], however, here *we propose an entirely different approach which does not require making additional variational approximations* (see Appendix G for further discussion).

Local Slot Mixtures. Probabilistic slot attention augments standard slot attention by introducing a per-datapoint (i.e. local) Gaussian Mixture Model (GMM) for learning slot distributions. Intuitively, a local GMM can be understood as a way to cluster features within a given image, encouraging the grouping of similar features into object representations. However, unlike regular clustering, here the clustered points are dynamically transformed representations of the actual data. Specifically,

[2] E.g. for a CNN with 500K parameters with batch size 32, $\geq$ 125GB of GPU memory is needed

[3] Note that we may use $\mathbf{s}^t$ or $\mathbf{s}(t)$ interchangeably to denote slot representations at slot attention iteration t.

we use an encoder function f_e that maps each image $\mathbf{x}_i \in \mathbb{R}^{H \times W \times C}$ in the dataset $\{\mathbf{x}_i\}_{i=1}^M$, to a latent spatial representation $\mathbf{z}_i \in \mathbb{R}^{N \times d}$. The latent variable $\mathbf{z}$ may be deterministic or stochastic, and we consider the case where $N < HW$ to reflect a modest downscaling with respect to (w.r.t.) $\mathbf{x}$. The goal is to dynamically map each of the N, d-dimensional vector representations in each $\mathbf{z}$, to one-of-K object slot distributions within a mixture. A *local* GMM can be fit to each posterior latent representation $\mathbf{z} \sim q(\mathbf{z} \mid \mathbf{x}_i)$[4] on the fly by maximizing likelihood:

$$p(\mathbf{z} \mid \boldsymbol{\pi}_i, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i) = \prod_{n=1}^{N} \sum_{k=1}^{K} \boldsymbol{\pi}_{ik} \mathcal{N}(\mathbf{z}_n; \boldsymbol{\mu}_{ik}, \boldsymbol{\sigma}_{ik}^2), \tag{2}$$

where $\boldsymbol{\mu}_i = (\boldsymbol{\mu}_{i1}, \ldots, \boldsymbol{\mu}_{iK})$, $\boldsymbol{\sigma}_i^2 = (\boldsymbol{\sigma}_{i1}^2, \ldots, \boldsymbol{\sigma}_{iK}^2)$ and $\boldsymbol{\pi}_i = (\pi_{i1}, \ldots, \pi_{iK})$ are the respective means, diagonal covariances and mixing coefficients of the i^{th} K-component mixture. Figure 2b illustrates the resulting probabilistic graphical model (PGM) in more detail.

To maximize the likelihood in Equation 2 per datapoint $\mathbf{x}_i$, we present a bespoke expectation-maximisation (EM) algorithm for slot attention, yielding closed-form update equations for the parameters as shown in Algorithm 1, and explained next.

Algorithm 1 Probabilistic Slot Attention

Input: $\mathbf{z} = f_e(\mathbf{x}) \in \mathbb{R}^{N \times d}$ ▷ representation
$\mathbf{k} \leftarrow \boldsymbol{W}_k \mathbf{z} \in \mathbb{R}^{N \times d}$ ▷ compute keys
$\mathbf{v} \leftarrow \boldsymbol{W}_v \mathbf{z} \in \mathbb{R}^{N \times d}$ ▷ optional $\mathbf{v} \coloneqq \mathbf{k}$
$\forall k, \boldsymbol{\pi}(0)_k \leftarrow \frac{1}{K}, \boldsymbol{\mu}(0)_k \sim \mathcal{N}(0, \mathbf{I}_d), \boldsymbol{\sigma}(0)_k^2 \leftarrow \mathbf{1}_d$
for $t = 0, \ldots, T-1$
$A_{nk} \leftarrow \dfrac{\boldsymbol{\pi}(t)_k \mathcal{N}\left(\mathbf{k}_n; \boldsymbol{W}_q \boldsymbol{\mu}(t)_k, \boldsymbol{\sigma}(t)_k^2\right)}{\sum_{j=1}^{K} \boldsymbol{\pi}(t)_j \mathcal{N}\left(\mathbf{k}_n; \boldsymbol{W}_q \boldsymbol{\mu}(t)_j, \boldsymbol{\sigma}(t)_j^2\right)}$
$\hat{A}_{nk} \leftarrow A_{nk} / \sum_{l=1}^{N} A_{lk}$ ▷ normalize
$\boldsymbol{\mu}(t+1)_k \leftarrow \sum_{n=1}^{N} \hat{A}_{nk} \mathbf{v}_n$ ▷ update slots
$\boldsymbol{\sigma}(t+1)_k^2 \leftarrow \sum_{n=1}^{N} \hat{A}_{nk} \left(\mathbf{v}_n - \boldsymbol{\mu}(t+1)_k\right)^2$
$\boldsymbol{\pi}(t+1)_k \leftarrow \sum_{n=1}^{N} A_{nk} / N$ ▷ update mixing
return $\boldsymbol{\mu}(T), \boldsymbol{\sigma}(T)^2$ ▷ K slot distributions

Probabilistic Projections. A powerful property of slot attention and cross-attention more broadly [74], is its ability to decouple the *agreement* mechanism from the representational content. That is, the dot-product is used to measure agreement between each *query* (slot) vector and all the *key* vectors, to dictate how much of each *value* vector (content from $\mathbf{z}_i$) should be represented in each slot's revised representation. To retain this flexibility and decouple the attention computation from the content, we incorporate key-value projections into our probabilistic approach. For brevity, the i subscript is implicit in the following, keeping in mind that these are local quantities (per-datapoint $\mathbf{x}_i$). The parameters of the K Gaussian slot distributions are initialized (at attention iteration $t = 0$) as follows:

$$\forall k, \qquad \boldsymbol{\pi}(0)_k = K^{-1}, \qquad \boldsymbol{\mu}(0)_k \sim \mathcal{N}(0, \mathbf{I}_d), \qquad \boldsymbol{\sigma}(0)_k^2 = \mathbf{1}_d. \tag{3}$$

The respective queries $\mathbf{q}$, keys $\mathbf{k}$, and values $\mathbf{v}$ are then given by:

$$\mathbf{q}(t) = \boldsymbol{W}_q \boldsymbol{\mu}(t), \qquad \mathbf{k} = \boldsymbol{W}_k \mathbf{z}, \qquad \mathbf{v} = \boldsymbol{W}_v \mathbf{z}, \tag{4}$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d \times d}$, whereas $\mathbf{q}(t)$ denotes the queries at attention iteration t. To measure agreement between each input feature (key) and slot (query), we evaluate the normalized probability density of each key under a Gaussian model defined by each slot:

$$A_{nk} = \frac{1}{Z} \boldsymbol{\pi}(t)_k \mathcal{N}\left(\mathbf{k}_n; \mathbf{q}(t)_k, \boldsymbol{\sigma}(t)_k^2\right), \qquad Z = \sum_{j=1}^{K} \boldsymbol{\pi}(t)_j \mathcal{N}\left(\mathbf{k}_n; \mathbf{q}(t)_k, \boldsymbol{\sigma}(t)_j^2\right), \tag{5}$$

where A_{nk} corresponds to the posterior probability that slot (query) k is responsible for input feature (key) n. This process yields the slot attention matrix $\boldsymbol{A} \in \mathbb{R}^{N \times K}$. As shown in Algorithm 1, the mixture parameters $\boldsymbol{\pi}(t), \boldsymbol{\mu}(t), \boldsymbol{\sigma}(t)^2$ are then updated using the attention matrix and the values $\mathbf{v}$. If the values are chosen to be equal to the keys $\mathbf{v} \coloneqq \mathbf{k}$, then the procedure is more in line with standard EM, but the agreement mechanism and the content become entangled. After T probabilistic slot attention iterations, the resulting K Gaussians serve as slot posterior distributions:

$$\mathbf{s}(T)_k \sim \mathcal{N}\left(\boldsymbol{\mu}(T)_k, \boldsymbol{\sigma}(T)_k^2\right), \qquad \text{for } k = 1, \ldots, K, \tag{6}$$

where $\boldsymbol{\mu}(T)$ and $\boldsymbol{\sigma}(T)^2$ are the parameters of all the Gaussians in the mixture given a particular datapoint $\mathbf{x}$. The slots $\mathbf{s}(T)_{1:K}$ are then used for input reconstruction, e.g. by maximizing a (possibly Gaussian) likelihood $p(\mathbf{x} \mid \mathbf{s}(T)_{1:K})$ parameterized by a (possibly additive) decoder f_d.

[4] The parametric form of q can be e.g. Gaussian or Dirac delta.

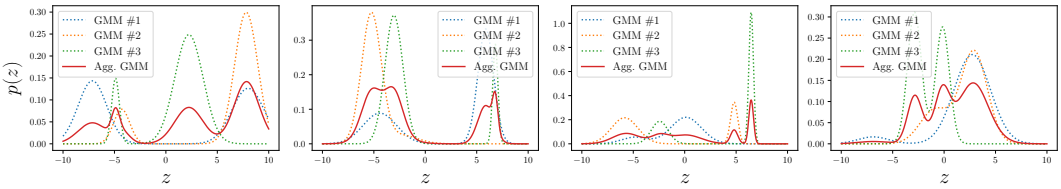


Figure 3: **Aggregate Gaussian Mixture Density.** Examples of aggregate posterior mixtures. For each plot, we compute the aggregate mixture (red line) based on three random bimodal Gaussian mixtures, and plot the respective densities. The three GMMs here are analogous to the local GMMs obtained from probabilistic slot attention (Algorithm 1), and the aggregate GMM represents $q(\mathbf{z})$.

Computational Complexity. Probabilistic slot attention (PSA) retains the $\mathcal{O}(TNKd)$ computational complexity of slot attention. The additional operations we introduce for calculating slot mixing coefficients and slot variances (under diagonal slot covariance structure) have complexities of $\mathcal{O}(NK)$ and $\mathcal{O}(NKd)$ respectively, which do not alter the dominant term. When PSA is combined with an additive decoder, it can *lower* computational complexity by eliminating the need to decode inactive slots. In the following, we outline a principled approach for pruning inactive slots.

Automatic Relevance Determination of Slots. An open problem in slot-based modelling is the dynamic estimation of the number of slots K needed for each input [51, 44]. Probabilistic slot attention offers an elegant solution to this problem using the concept of Automatic Relevance Determination (ARD) [60]. ARD is a statistical framework that prunes irrelevant features by imposing data-driven, sparsity-inducing priors on model parameters to regularize the solution space. Since the output mixing coefficients $\boldsymbol{\pi}(T) \in \mathbb{R}^K$ are input dependent (i.e. local), irrelevant components (slots) will naturally be pruned after T attention iterations, i.e. $\boldsymbol{\pi}(T)_k \to 0$ for any unused slot k. We can either use a probability threshold $\tau \in [0, 1)$ to prune unused slots or place a Dirichlet prior over the mixing coefficients to explicitly induce sparsity. For simplicity, we take the former approach:

$$\mathbf{s}_\tau \coloneqq \left\{\mathbf{s}(T)_k \mid k \in [K], \boldsymbol{\pi}(T)_k > \tau\right\}, \tag{7}$$

where $\mathbf{s}_\tau$ denotes the set of *active* slots with mixing coefficient greater than τ, and each slot is (optionally) sampled from its Gaussian distribution: $\mathbf{s}(T)_k \sim \mathcal{N}\left(\boldsymbol{\mu}(T)_k, \boldsymbol{\sigma}(T)_k^2\right)$.

Aggregate Posterior Gaussian Mixture. As previously explained, probabilistic slot attention goes beyond standard slot attention by introducing a per-datapoint (i.e. local) GMM to learn *distributions* over slot representations. This imposes structure over the latent space and gives us access to *posterior* slot distributions after the attention iterations. Rather than constraining slot posteriors to be close to a tractable prior – e.g. via the VAE framework [41, 76] which requires further variational approximations – we leverage our probabilistic setup to compute the optimal (global) prior over slots.

We propose to compute the *aggregate slot posterior* by marginalizing out the data: $q(\mathbf{z}) = \sum_{i=1}^{M} q(\mathbf{z} \mid \mathbf{x}_i)/M$, given a pre-trained probabilistic slot attention model (Fig. 3). In § 5, we prove that the aggregate posterior is a tractable, non-degenerate Gaussian mixture distribution which: **(i)** Serves as the theoretically optimal prior over slots; **(ii)** Is empirically stable across runs (i.e. identifiable up to an affine transformation and slot permutation, § 5); **(iii)** Can be tractably sampled from and (optionally) used for slot-based scene composition tasks. Since GMMs are universal density approximators given enough components (even GMMs with diagonal covariances), the resulting aggregate posterior $q(\mathbf{z})$ is highly flexible and multimodal. It often suffices to approximate it using a sufficiently large subset of the dataset, if marginalizing out the entire dataset becomes computationally restrictive, although we did not observe this to be the case in practice in our set of experiments.

5 Theory: Slot Identifiability Result

In this section, we leverage the properties of the probabilistic slot attention method proposed in Section 4 to prove a new object-centric identifiability result. We show that object-centric representations (i.e. slots) are identifiable without supervision (up to an equivalence relation) under mixture model-like assumptions about the latent space. This contrasts with existing work, which provides identifiability guarantees within a specific class of mixing functions, i.e. additive decoders [47]. Our result unifies generative model identifiablity [32, 37, 43] and object-centric learning.

Definition 2 (Identifiability.)**.** Given an observation space $\mathcal{X}$, a probabilistic model p with parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is said to be identifiable if the mapping $\boldsymbol{\theta} \in \boldsymbol{\Theta} \mapsto p_{\boldsymbol{\theta}}(\mathbf{x})$ is injective:

$$(p_{\boldsymbol{\theta}_1}(\mathbf{x}) = p_{\boldsymbol{\theta}_2}(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}) \implies \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2. \tag{8}$$

Remark 1. Definition 2 says that if any two choices of model parameters lead to the same marginal density, they are equal. This is often referred to as *strong* identifiability [32, 37], and it can be too restrictive, as guaranteeing identifiability up to a simple transformation (e.g. affine) is acceptable in practice. To reflect weaker notions of identifiability, we let $\sim$ denote an equivalence relation on $\boldsymbol{\Theta}$, such that a model can be said to be identifiable up to $\sim$, or $\sim$-identifiable.

Definition 3 ($\sim_s$-equivalence)**.** Let $f_{\boldsymbol{\theta}} : \mathcal{S} \to \mathcal{X}$ denote a mapping from slot representation space $\mathcal{S}$ to image space $\mathcal{X}$ (satisfying Assumption 8), the equivalence relation $\sim_s$ w.r.t. to parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is defined bellow, where $\boldsymbol{P} \in \mathcal{P} \subseteq \{0, 1\}^{K \times K}$ is a slot permutation matrix, $\boldsymbol{H} \in \mathbb{R}^{d \times d}$ is an affine transformation matrix, and $\mathbf{c} \in \mathbb{R}^d$:

$$\forall \mathbf{x} \in \mathcal{X}, \qquad \boldsymbol{\theta}_1 \sim_s \boldsymbol{\theta}_2 \iff \exists\, \boldsymbol{P}, \boldsymbol{H}, \mathbf{c} : f_{\boldsymbol{\theta}_1}^{-1}(\mathbf{x}) = \boldsymbol{P}(f_{\boldsymbol{\theta}_2}^{-1}(\mathbf{x})\boldsymbol{H} + \mathbf{c}). \tag{9}$$

Lemma 1 (Aggregate Posterior Mixture)**.** *Given that probabilistic slot attention induces a local (per-datapoint $\mathbf{x} \in \{\mathbf{x}_i\}_{i=1}^M$) GMM with K components, the aggregate posterior $q(\mathbf{z})$ obtained by marginalizing out $\mathbf{x}$ is a non-degenerate global Gaussian mixture with MK components:*

$$q(\mathbf{z}) = \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} \widehat{\boldsymbol{\pi}}_{ik} \mathcal{N}\left(\mathbf{z}; \widehat{\boldsymbol{\mu}}_{ik}, \widehat{\boldsymbol{\sigma}}_{ik}^2\right). \tag{10}$$

Proof Sketch. The full proof is given in Appendix B. The result is obtained by integrating the product of the latent variable posterior density $q(\mathbf{z} \mid \mathbf{x})$ and the (local) GMM density given $\mathbf{z}$, w.r.t. $\mathbf{x}$. We then proceed by verifying that the mixing coefficients sum to one over all the components in the new mixture (Corollary 4), proving aggregated posterior to be a well-defined probability distribution, this can be empirically confirmed in Figure 3. Next, we use $q(\mathbf{z})$ in our identifiablity result.

Theorem 1 (Mixture Distribution of Concatenated Slots)**.** *Let f_s denote a permutation equivariant PSA function such that $f_s(\mathbf{z}, P\mathbf{s}^t) = Pf_s(\mathbf{z}, \mathbf{s}^t)$, where $P \in \{0, 1\}^{K \times K}$ is an arbitrary permutation matrix. Let $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_K) \in \mathbb{R}^{Kd}$ be a random variable defined as the concatenation of K individual slots, where each slot is Gaussian distributed within a K-component mixture: $\mathbf{s}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \in \mathbb{R}^d, \forall k \in \{1, \ldots K\}$. Then, $\mathbf{s}$ is also GMM distributed with $K!$ mixture components:*

$$p(\mathbf{s}) = \sum_{p=1}^{K!} \boldsymbol{\pi}_p \mathcal{N}(\mathbf{s}; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p), \quad \textit{where} \quad \boldsymbol{\pi} \in \Delta^{K!-1}, \quad \boldsymbol{\mu}_p \in \mathbb{R}^{Kd}, \quad \boldsymbol{\Sigma}_p \in \mathbb{R}^{Kd \times Kd}. \tag{11}$$

Proof Sketch. The proof is in Appendix B. We observe that the permutation equivariance of the PSA function f_s induces $K!$ ways of concatenating sampled slots $\mathbf{s}_k$, where each permutation maps to a different mode with block diagonal covariance structure in a GMM living in $\mathbb{R}^{Kd}$ (e.g. Fig. 6, 7).

Theorem 2 ($\sim_s$-Identifiable Slot Representations)**.** *Given that the aggregate posterior $q(\mathbf{z})$ is an optimal, non-degenerate mixture prior over slot space (Lemma 1), $f : \mathcal{S} \to \mathcal{X}$ is a piecewise affine weakly injective mixing function (Assumption 8), and the slot representation, $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_K)$ can be observed as a sample from a GMM (Theorem 1), then $p(\mathbf{s})$ is identifiable as per Definition 3.*

Proof Sketch. The proof is given in Appendix D. Lemma 1 and Corollary 4 state that the optimal latent variable prior in our case is GMM distributed, non-degenerate and equates to the aggregate posterior $q(\mathbf{z})$. This permits us to extend [43]'s result to show that if $q(\mathbf{z})$ is distributed according to a non-degenerate GMM and the mixing function f_d is piecewise affine and weakly injective, then the *slot* distribution representation, $p(\mathbf{s})$ which is also a GMM (Theorem 1) is identifiable up to an affine transformation and arbitrary slot permutation.

Corollary 3 (Individual Slot Identifiability)**.** *If the distribution over concatenated slots $p(\mathbf{s})$, where $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_K) \in \mathbb{R}^{Kd}$, is $\sim_s$-identifiable (Theorem 2) then this implies $q(\mathbf{z})$ is identifiable up to an affine transformation and permutation of the slots $\mathbf{s}_k$. Therefore, each slot distribution $\mathbf{s}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \in \mathbb{R}^d, \forall k \in \{1, \ldots K\}$ is also identifiable up to an affine transformation.*

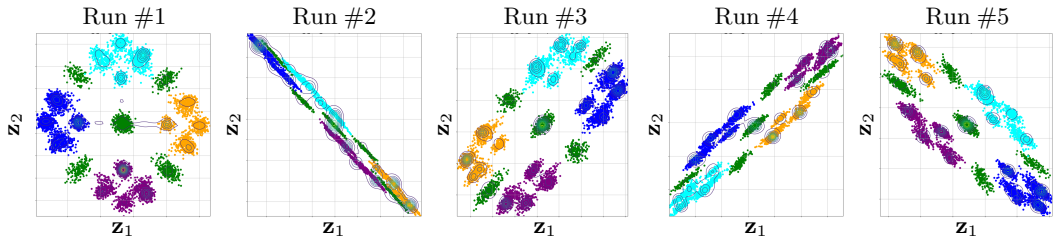


Figure 4: **Aggregate posterior identifiability**. Recovered (latent) aggregate posteriors $q(\mathbf{z})$ across 5 runs of our PSA model. As detailed in Section 6, we used a 2D synthetic dataset with 5 total 'object' clusters, with each observation containing at most 3. This provides strong evidence of recovery of the latent space up to affine transformations, empirically verifying our identifiability claim.

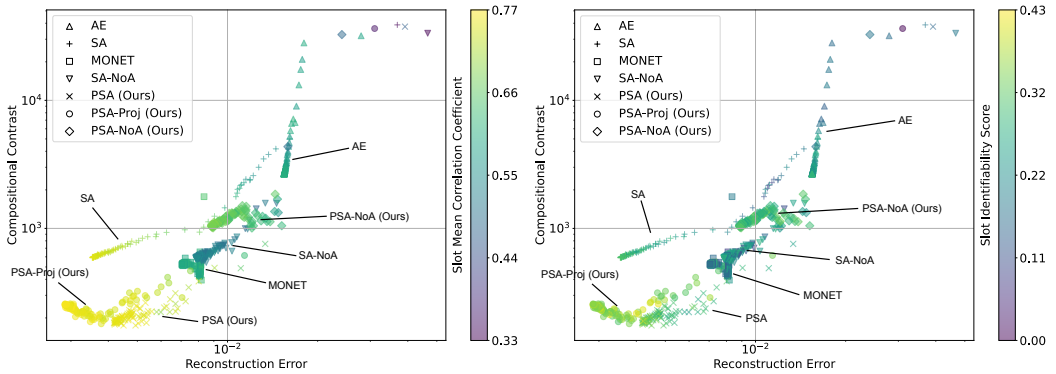


Figure 5: **Experiments comparing slot-identifiability scores**. Colour coding represents the level of slot identifiability achieved by each model as measured by the SMCC (left) and SIS (right). SMCC offers a more consistent metric correlating with reconstruction loss.

6 Experiments

Given that the focus of this work is theoretical, the primary goal of our experiments is to provide strong empirical evidence of our main identifiability result (ref. Figures 4, 5). With that said, we also extend our experimental study to popular imaging benchmarks to demonstrate that our method scales to higher-dimensional settings (ref. Tables 2, 4).

Datasets & Evaluation Metrics. Our experimental analysis involves standard benchmark datasets from object-centric learning literature including SPRITEWORLD [6], CLEVR [34], and OBJECT-SROOM [35]. We report the foreground-adjusted rand index (FG-ARI) and FID [23] to quantify both object-level *binding* capabilities and image quality. Our main goal is to measure slot-identifiability, so we use the slot identifiability score [6] and the mean correlation coefficient (MCC) across slot representations – we call the latter slot-MCC (SMCC). For two sets of slots $\{\mathbf{s}_i\}_{i=1}^M$, and $\{\tilde{\mathbf{s}}_i\}_{i=1}^M$, where $\mathbf{s}_i \in \mathbb{R}^{K\times d}$, $\tilde{\mathbf{s}}_i \in \mathbb{R}^{K\times d}$, extracted from M images $\{\mathbf{x}_i\}_{i=1}^M$, the SMCC between any $\mathbf{s}$ and $\tilde{\mathbf{s}}$ is obtained by matching the slot representations and their order. The order is matched by mapping slots in $\tilde{\mathbf{s}}$ w.r.t $\mathbf{s}$ assigned by π, followed by a learned affine mapping $\boldsymbol{A}$ between aligned $\tilde{\mathbf{s}}_{\pi(i)}$ and $\mathbf{s}$:

$$\text{SMCC}(\mathbf{s}, \tilde{\mathbf{s}}) := \frac{1}{K \times d} \sum_{i=0}^{K} \sum_{j=0}^{d} \rho(\mathbf{s}_{ij}, \boldsymbol{A}\tilde{\mathbf{s}}_{\pi(i)j}). \tag{12}$$

For more details on the metrics please refer to Appendix F.

Models & Baselines. We consider three ablations on our proposed probabilistic slot attention (PSA) method: (i) PSA base model (Algorithm 2); (ii) PSA-PROJ model (Algorithm 1); and (iii) PSA-NOA model, which is equivalent to PSA-PROJ but without an additive decoder. We experiment with two types of decoders: (i) an additive decoder similar to [79]'s spatial broadcasting model; and (ii) standard convolutional decoder. In all cases, we use LeakyReLU activations to satisfy the weak

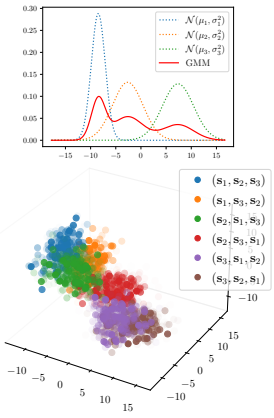


Table 2: Comparing slot identifiability scores (SMCC and slot averaged R2) with existing object-centric learning methods.

METHOD	CLEVR		OBJECTS-ROOM	
	SMCC ↑	R2 ↑	SMCC ↑	R2 ↑
AE	$0.43 \pm .02$	$0.26 \pm .02$	$0.46 \pm .05$	$0.45 \pm .06$
MONET	$0.32 \pm .01$	$0.39 \pm .09$	$0.43 \pm .04$	$0.41 \pm .10$
SA	$0.56 \pm .02$	$0.55 \pm .05$	$0.66 \pm .01$	$0.54 \pm .00$
SA-NoA	$0.23 \pm .03$	$0.24 \pm .02$	$0.48 \pm .02$	$0.47 \pm .01$
Ours:				
PSA-NOA	$0.56 \pm .05$	$0.42 \pm .06$	$0.44 \pm .03$	$0.45 \pm .05$
PSA	$0.58 \pm .06$	$0.48 \pm .02$	$0.66 \pm .01$	$\mathbf{0.64 \pm .02}$
PSA-PROJ	$\mathbf{0.66 \pm .06}$	$\mathbf{0.62 \pm .08}$	$\mathbf{0.71 \pm .00}$	$0.62 \pm .02$

Figure 6: **Concatenated Slot Gaussian Mixture**. Example of the higher-dim GMM (1D $\rightarrow$ 3D) induced by the permutation equivariance of PSA and the $K!$ ways of concatenating sampled slots.

injectivity conditions (Assumption 8). In terms of object-centric learning baselines, we compare with standard additive autoencoder setups following [6], slot-attention (SA) [53], and MONET [9].

Verifying Slot Identifiability: Gaussian Mixture of Objects. To provide conclusive empirical evidence of our identifiability claim (Theorem 2), we set up a synthetic modelling scenario under which it is possible to visualize the aggregate posterior across different runs. The goal is to show that PSA is $\sim_s$-identifiable (Theorem 2) in the sense that it can recover the same latent space distribution up to an affine transformation and slot order permutation. For the data generating process, we defined a $K{=}5$ component GMM, with differing mean parameters $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_5\}$, and shared isotropic covariances. The 5 components emulate 5 different object types in a given environment. To create a single data point, we randomly chose 3 of the 5 components and sampled 128 points uniformly at random from each mode. Figure 9 shows some data samples, where different colours correspond to different objects. We used 1000 data points in total for training our PSA model. As shown in Figure 4, the aggregate posterior is either rotated, translated, skewed, or flipped across different runs as predicted by our theory – this contrasts with all baselines wherein the aggregate posterior is intractable. We observed an SMCC of 0.93 ± 0.04, and R2-score of 0.50 ± 0.08.

Case Study: Imaging Data. Although our focus is primarily theoretical, we now demonstrate that our method generalizes/scales to higher-dimensional imaging modalities. To that end, we first use the SPRITEWORLD [78] dataset to evaluate the SMCC and SIS w.r.t. ground truth latent variables. Figure 5 presents our identifiability results against the baselines. Similar to [6], we observe higher SIS when compositional contrast and reconstruction error decreases. However, when the mixing function is *not* additive, the compositional contrast does not decrease drastically while maintaining higher SMCC and SIS – this verifies our identifiability claim using only piecewise affine decoders. As shown in Figure 5, we also observe that PSA routing with additive decoder models achieves lower compositional contrast and reconstruction errors when compared with other methods. This indicates that stronger identifiability of slot representations is achievable when combining both slot latent structure and inductive biases in the mixing function. Unlike for the SPRITEWORLD dataset, all the ground truth generative factors are *unobserved* for the CLEVR and OBJECTSROOM datasets we use. Therefore, for evaluation in these cases, we train multiple models with different seeds and compute the SMCC and SIS measures across different runs. This is similar to our earlier synthetic experiment and standard practice in identifiability literature. Table 2 presents our main identifiability results on CLEVR and OBJECTSROOM. We observe similar trends in favour of our proposed PSA method as measured by both SMCC and (slot averaged) R2 score relative to the baselines.

Case Study: Complex Decoder Structure. To empirically test slot identifiability using more complex non-additive decoders, we used transformer decoders following SLATE [67], and simply replaced the slot attention module with probabilistic slot attention. On the CLEVR dataset, we observed a significantly improved SMCC of $\mathbf{0.73} \pm .01$ and R2 of $\mathbf{0.55} \pm .06$ relative to Table 2.

To demonstrate that probabilistic slot attention can scale to large-scale real-world data we ran additional experiments on the Pascal VOC2012 [49] dataset, following the exact DINOSAUR strategies and setups described in [66, 36] for fairness, then simply swapping out the slot attention module with probabilistic slot attention. As shown in Table 3, we find that probabilistic slot attention is competitive with standard slot attention on real-world data. We also tested more complex, *non-additive* decoders based on autoregressive transformers, following the DINOSAUR [66] setup. For our PSA TRANSFORMER (W/ DINO) model, we observed an MBO_i of **0.447**, and MBO_c of **0.521** which is competitive with the DINOSAUR TRANSFORMER baseline [66] of 0.44 and 0.512 respectively. In this case, we found that a lower maximum learning rate of 10^{-4} was beneficial for stabilizing PSA training. In summary, our experiments corroborate our theoretical results and suggest why *non-additive* decoder structures can still work well given the appropriate latent structure and inference procedures are in place. With that said, there is a trade-off between identifiability and expressivity induced by the choice of decoder structure [47], so depending on the use case, it may indeed be advantageous to combine both latent and additive decoder structures in practice.

Table 3: Pascal VOC2012 benchmark results using probabilistic slot attention (PSA). All baselines are standard results from [54]. SA MLP (W/ DINO) denotes our replication of the DINOSAUR MLP baseline from [66], whereas (‡) denotes the use of slot attention masks rather than decoder alpha masks for evaluation.

MODEL	$\text{MBO}_i\uparrow$	$\text{MBO}_c\uparrow$
BLOCK MASKS	$0.247 \pm .000$	$0.259 \pm .000$
SA	$0.222 \pm .008$	$0.237 \pm .008$
SLATE	$0.310 \pm .004$	$0.324 \pm .004$
ROTATING FEATURES	$0.282 \pm .006$	$0.320 \pm .006$
DINO K-MEANS	$0.363 \pm .000$	$0.405 \pm .000$
DINO CAE	$0.329 \pm .009$	$0.374 \pm .010$
DINOSAUR MLP	$0.395 \pm .000$	$0.409 \pm .000$
Ours:		
SA MLP (W/ DINO)	$0.384 \pm .000$	$0.397 \pm .000$
SA MLP (W/ DINO)‡	$0.400 \pm .000$	$0.415 \pm .000$
PSA MLP (W/ DINO)	$0.389 \pm .009$	$0.422 \pm .009$
PSA MLP (W/ DINO)‡	$\mathbf{0.405} \pm .010$	$\mathbf{0.436} \pm .011$

7 Discussion

Understanding when object-centric representations can theoretically be identified is important for scaling slot-based methods to high-dimensional images with correctness guarantees. In contrast with existing work, which focuses primarily on properties of the slot *mixing function*, we leverage distributional assumptions about the slot latent space to prove a new slot-identifiability result. Specifically, we prove that object-centric representations are identifiable without supervision (up to an equivalence relation) under mixture model-like distributional assumptions on the latent slots. To that end, we proposed a probabilistic slot-attention algorithm that imposes an *aggregate* mixture prior over slot representations which is both demonstrably stable across runs and tractable to sample from. Our empirical study primarily verifies our theoretical identifiability claim and demonstrates that our framework achieves the lowest compositional contrast without being explicitly trained towards that objective, which is computationally infeasible beyond toy datasets. In summary, we show how slot identifiability can be achieved via probabilistic constraints on the latent space and piecewise decoders. These piecewise decoders manifest as e.g. standard MLPs with LeakyReLU activations and are generally less restrictive than additive decoders. When coupling probabilistic and additive decoder structures, we observe further performance improvements relative to either one in isolation.

Limitations & Future Work. We recognize that our assumptions, particularly the *weak injectivity* of the mixing function, may not always hold in practice for different types of architectures (see Appendix C for sufficiency conditions). Although generally applicable, the piecewise-affine functions we use may not always accurately reflect valid assumptions about real-world problems, *e.g.* when the model is misspecified. Like all object-centric learning methods, we also assume that the mixing function is invariant to permutations of the slots in practice, which technically makes it non-invertible. We deem this aspect to be an interesting area for future work, as an extension to accommodate permutation invariance would strengthen and generalize the identifiability guarantees we provide. Additionally, we do not study cases where objects are occluded, *i.e.* when are shared or bordering other objects. This limitation is not unique to our work [53, 6, 14, 15, 44] and overcoming it requires further investigation by the community. Nonetheless, our theoretical results capture the important concepts in object-centric learning and represent a valuable extension to the nascent theoretical foundations of the area. In future work, it would be valuable to further relax slot identifiability requirements/assumptions and study slot compositional properties of probabilistic slot attention.

Acknowledgements

A. Kori is supported by UKRI (grant number EP/S023356/1), as part of the UKRI Centre for Doctoral Training in Safe and Trusted AI. B. Glocker and F.D.S. Ribeiro acknowledge the support of the UKRI AI programme, and the Engineering and Physical Sciences Research Council, for CHAI - EPSRC Causality in Healthcare AI Hub (grant number EP/Y028856/1).

References

[1] Kartik Ahuja, Jason S Hartford, and Yoshua Bengio. Weakly supervised representation learning with sparse perturbations. *Advances in Neural Information Processing Systems*, 35:15516–15528, 2022. [Cited on page 3.]

[2] Kartik Ahuja, Yixin Wang, Divyat Mahajan, and Yoshua Bengio. Interventional causal representation learning. *arXiv preprint arXiv:2209.11924*, 2022. [Cited on page 2.]

[3] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. [Cited on page 1.]

[4] Timothy EJ Behrens, Timothy H Muller, James CR Whittington, Shirley Mark, Alon B Baram, Kimberly L Stachenfeld, and Zeb Kurth-Nelson. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron*, 100(2):490–509, 2018. [Cited on page 1.]

[5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. [Cited on pages 1 and 2.]

[6] Jack Brady, Roland S Zimmermann, Yash Sharma, Bernhard Schölkopf, Julius von Kügelgen, and Wieland Brendel. Provably learning object-centric representations. *arXiv preprint arXiv:2305.14229*, 2023. [Cited on pages 1, 2, 3, 4, 8, 9, 10, 17, 21, and 24.]

[7] Johann Brehmer, Pim De Haan, Phillip Lippe, and Taco S Cohen. Weakly supervised causal representation learning. *Advances in Neural Information Processing Systems*, 35:38319–38331, 2022. [Cited on page 2.]

[8] Simon Buchholz, Michel Besserve, and Bernhard Schölkopf. Function classes for identifiable nonlinear independent component analysis. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. [Cited on page 2.]

[9] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019. [Cited on pages 2 and 9.]

[10] Michael Chang, Thomas L Griffiths, and Sergey Levine. Object representations as fixed points: Training iterative refinement algorithms with implicit differentiation. *arXiv preprint arXiv:2207.00787*, 2022. [Cited on page 1.]

[11] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016. [Cited on pages 2 and 25.]

[12] Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. [Cited on page 2.]

[13] Gamaleldin Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C Mozer, and Thomas Kipf. Savi++: Towards end-to-end object-centric learning from real-world videos. *Advances in Neural Information Processing Systems*, 35:28940–28954, 2022. [Cited on page 2.]

[14] Patrick Emami, Pan He, Sanjay Ranka, and Anand Rangarajan. Slot order matters for compositional scene understanding. *arXiv preprint arXiv:2206.01370*, 2022. [Cited on pages 2 and 10.]

[15] Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv preprint arXiv:1907.13052*, 2019. [Cited on pages 1, 2, and 10.]

[16] Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. Genesis-v2: Inferring unordered object representations without iterative refinement. *Advances in Neural Information Processing Systems*, 34:8085–8094, 2021. [Cited on pages 1 and 2.]

[17] Russell A Epstein, Eva Zita Patai, Joshua B Julian, and Hugo J Spiers. The cognitive map in humans: spatial navigation and beyond. *Nature neuroscience*, 20(11):1504–1513, 2017. [Cited on page 1.]

[18] Fabian Falck, Haoting Zhang, Matthew Willetts, George Nicholson, Christopher Yau, and Chris C Holmes. Multi-facet clustering variational autoencoders. *Advances in Neural Information Processing Systems*, 34:8676–8690, 2021. [Cited on page 2.]

[19] Tobias Gerstenberg, Noah D Goodman, David A Lagnado, and Joshua B Tenenbaum. A counterfactual simulation model of causal judgments for physical events. *Psychological review*, 128(5):936, 2021. [Cited on page 1.]

[20] Alison Gopnik, Clark Glymour, David M Sobel, Laura E Schulz, Tamar Kushnir, and David Danks. A theory of causal learning in children: causal maps and bayes nets. *Psychological review*, 111(1):3, 2004. [Cited on page 1.]

[21] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, pages 2424–2433. PMLR, 2019. [Cited on page 2.]

[22] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020. [Cited on pages 1 and 3.]

[23] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. [Cited on page 8.]

[24] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. [Cited on page 2.]

[25] Geoffrey Hinton. Some demonstrations of the effects of structural descriptions in mental imagery. *Cognitive Science*, 3(3):231–250, 1979. [Cited on page 1.]

[26] Geoffrey Hinton. How to represent part-whole hierarchies in a neural network. *Neural Computation*, pages 1–40, 2022. [Cited on page 3.]

[27] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *International conference on learning representations*, 2018. [Cited on page 3.]

[28] Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1, 2016. [Cited on pages 18 and 25.]

[29] A Hyvarinen and E Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000. [Cited on pages 1 and 2.]

[30] Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *Advances in neural information processing systems*, 29, 2016. [Cited on page 2.]

[31] Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999. [Cited on pages 1 and 2.]

[32] Aapo Hyvarinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR, 2019. [Cited on pages 2, 6, and 7.]

[33] Aapo Hyvarinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89, pages 859–868. PMLR, 2019. [Cited on page 2.]

[34] 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*, pages 2901–2910, 2017. [Cited on page 8.]

[35] Rishabh Kabra, Chris Burgess, Loic Matthey, Raphael Lopez Kaufman, Klaus Greff, Malcolm Reynolds, and Alexander Lerchner. Multi-object datasets. https://github.com/deepmind/multi-object-datasets/, 2019. [Cited on page 8.]

[36] Ioannis Kakogeorgiou, Spyros Gidaris, Konstantinos Karantzalos, and Nikos Komodakis. Spot: Self-training with patch-order permutation for object-centric learning with autoregressive transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22776–22786, 2024. [Cited on page 10.]

[37] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020. [Cited on pages 2, 6, 7, 17, and 24.]

[38] Ilyes Khemakhem, Ricardo Monti, Diederik Kingma, and Aapo Hyvarinen. Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica. In *Advances in Neural Information Processing Systems*, volume 33, 2020. [Cited on pages 2, 17, and 24.]

[39] Ilyes Khemakhem, Ricardo Monti, Diederik Kingma, and Aapo Hyvarinen. Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica. *Advances in Neural Information Processing Systems*, 33:12768–12778, 2020. [Cited on page 24.]

[40] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2018. [Cited on page 2.]

[41] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. [Cited on pages 2 and 6.]

[42] Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*, 2021. [Cited on page 2.]

[43] Bohdan Kivva, Goutham Rajendran, Pradeep Ravikumar, and Bryon Aragam. Identifiability of deep generative models without auxiliary information. *Advances in Neural Information Processing Systems*, 35:15687–15701, 2022. [Cited on pages 2, 6, 7, 17, 21, and 22.]

[44] Avinash Kori, Francesco Locatello, Fabio De Sousa Ribeiro, Francesca Toni, and Ben Glocker. Grounded object centric learning. *arXiv preprint arXiv:2307.09437*, 2023. [Cited on pages 1, 2, 6, and 10.]

[45] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. *Advances in neural information processing systems*, 28, 2015. [Cited on page 1.]

[46] Sébastien Lachapelle, Pau Rodríguez López, Yash Sharma, Katie Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Nonparametric partial disentanglement via mechanism sparsity: Sparse actions, interventions and sparse temporal dependencies. *arXiv preprint arXiv:2401.04890*, 2024. [Cited on page 2.]

[47] Sébastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien. Additive decoders for latent variables identification and cartesian-product extrapolation. *Advances in Neural Information Processing Systems*, 36, 2024. [Cited on pages 1, 3, 4, 6, 10, and 17.]

[48] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017. [Cited on page 1.]

[49] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9215–9223, 2018. [Cited on page 10.]

[50] Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. *arXiv preprint arXiv:2001.02407*, 2020. [Cited on page 2.]

[51] 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*, pages 4114–4124. PMLR, 2019. [Cited on pages 1, 2, and 6.]

[52] Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 6348–6359. PMLR, 2020. [Cited on page 2.]

[53] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020. [Cited on pages 1, 2, 3, 4, 9, and 10.]

[54] Sindy Löwe, Phillip Lippe, Francesco Locatello, and Max Welling. Rotating features for object discovery. *Advances in Neural Information Processing Systems*, 36, 2024. [Cited on page 10.]

[55] Amin Mansouri, Jason Hartford, Kartik Ahuja, and Yoshua Bengio. Object-centric causal representation learning. In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022. [Cited on page 3.]

[56] Amin Mansouri, Jason Hartford, Yan Zhang, and Yoshua Bengio. Object-centric architectures enable efficient causal representation learning. *arXiv preprint arXiv:2310.19054*, 2023. [Cited on page 3.]

[57] Gary F Marcus. *The algebraic mind: Integrating connectionism and cognitive science*. MIT press, 2003. [Cited on page 1.]

[58] Joe Marino, Yisong Yue, and Stephan Mandt. Iterative amortized inference. In *International Conference on Machine Learning*, pages 3403–3412. PMLR, 2018. [Cited on page 2.]

[59] Emile Mathieu, Tom Rainforth, N Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. [Cited on page 2.]

[60] Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996. [Cited on page 6.]

[61] Fabio De Sousa Ribeiro, Kevin Duarte, Miles Everett, Georgios Leontidis, and Mubarak Shah. Learning with capsules: A survey. *arXiv preprint arXiv:2206.02664*, 2022. [Cited on page 3.]

[62] Fabio De Sousa Ribeiro, Georgios Leontidis, and Stefanos Kollias. Capsule routing via variational bayes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3749–3756, 2020. [Cited on page 3.]

[63] Fabio De Sousa Ribeiro, Georgios Leontidis, and Stefanos Kollias. Introducing routing uncertainty in capsule networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 6490–6502, 2020. [Cited on page 3.]

[64] Irvin Rock. Orientation and form. 1973. [Cited on page 1.]

[65] Bernhard Schölkopf and Julius von Kügelgen. From statistical to causal learning. *Proceedings of the International Congress of Mathematicians*, 2022. [Cited on page 1.]

[66] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, et al. Bridging the gap to real-world object-centric learning. *International Conference on Learning Representations*, 2023. [Cited on pages 1, 2, and 10.]

[67] Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate dall-e learns to compose. *International Conference on Learning Representations*, 2022. [Cited on pages 1, 2, and 9.]

[68] Gautam Singh, Yeongbin Kim, and Sungjin Ahn. Neural block-slot representations. *arXiv preprint arXiv:2211.01177*, 2022. [Cited on page 2.]

[69] Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. *Advances in Neural Information Processing Systems*, 35:18181–18196, 2022. [Cited on page 2.]

[70] Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285, 2011. [Cited on page 1.]

[71] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223. PMLR, 2018. [Cited on page 25.]

[72] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. [Cited on page 25.]

[73] Sjoerd Van Steenkiste, Karol Kurach, Jürgen Schmidhuber, and Sylvain Gelly. Investigating object compositionality in generative adversarial networks. *Neural Networks*, 130:309–325, 2020. [Cited on page 2.]

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. [Cited on pages 3 and 5.]

[75] Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *Advances in neural information processing systems*, 34:16451–16467, 2021. [Cited on page 2.]

[76] Yanbo Wang, Letao Liu, and Justin Dauwels. Slot-vae: Object-centric scene generation with slot attention. *arXiv preprint arXiv:2306.06997*, 2023. [Cited on pages 2 and 6.]

[77] Yanbo Wang, Letao Liu, and Justin Dauwels. Slot-VAE: Object-centric scene generation with slot attention. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 36020–36035. PMLR, 23–29 Jul 2023. [Cited on pages 4 and 25.]

[78] Nicholas Watters, Loic Matthey, Sebastian Borgeaud, Rishabh Kabra, and Alexander Lerchner. Spriteworld: A flexible, configurable reinforcement learning environment. https://github.com/deepmind/spriteworld/, 2019. [Cited on page 9.]

[79] Nicholas Watters, Loic Matthey, Christopher P Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *arXiv preprint arXiv:1901.07017*, 2019. [Cited on page 8.]

[80] Matthew Willetts and Brooks Paige. I don't need u: Identifiable non-linear ica without side information. *arXiv preprint arXiv:2106.05238*, 2021. [Cited on page 2.]

[81] Xiaojiang Yang, Yi Wang, Jiacheng Sun, Xing Zhang, Shifeng Zhang, Zhenguo Li, and Junchi Yan. Nonlinear ICA using volume-preserving transformations. In *International Conference on Learning Representations*, 2022. [Cited on page 2.]

[82] Dingling Yao, Danru Xu, Sébastien Lachapelle, Sara Magliacane, Perouz Taslakian, Georg Martius, Julius von Kügelgen, and Francesco Locatello. Multi-view causal representation learning with partial observability. 2024. [Cited on page 2.]

[83] Alan Yuille and Daniel Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 10(7):301–308, 2006. [Cited on page 1.]

[84] Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, pages 12979–12990. PMLR, 2021. [Cited on page 2.]

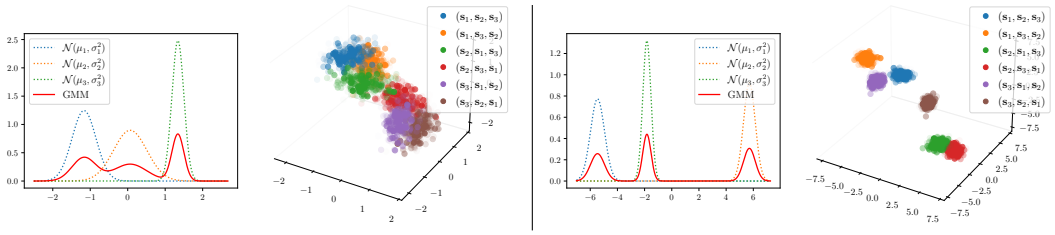


Figure 7: **Concatenated Slot Gaussian Mixture.** Examples of the higher dimensional Gaussian mixture induced by the permutation equivariance of slot attention and the $K!$ ways of concatenating sampled slots. Here we start with a random 1D GMM with $K = 3$ modes, each representing a different slot distribution, which then induces a respective 3D GMM with $K! = 6$ modes.

A List of Assumptions

Assumption 1 ($\mathcal{B}$- Disentanglement, [47])**.** Let $\mathbf{s} = \{\mathbf{s}_B, \forall\, B \in \mathcal{B}\}$ be a set of features wrt partition set $\mathcal{B}$. The learned mixing function f_d is said to be $\mathcal{B}$ disentangled wrt true decoder $\tilde{f}_d$ if there exists a permutation respecting diffeomorphism $v_B = \tilde{f}_d^{-1} \circ f_d \,\forall\, B \in \mathcal{B}$ which for a given feature $\mathbf{s}$ can be expressed as $v_B(\mathbf{s}) = v_B(\mathbf{s}_B)$.

Assumption 2 (Additive Mixing Function)**.** A mixing function $f_d : \mathcal{S} \to \mathcal{X}$, is said to be additive if there exist a partition set $\mathcal{B}$ and functions $f_d^B : s \to \mathbb{R}^{|\mathcal{X}|}$ such that: $f_d(\mathbf{s}) = \sum_{B \in \mathcal{B}} f_d^B(\mathbf{s}_B)$.

Assumption 3 (Irreducibility)**.** Given an an object $\mathbf{x}_k \in \mathbf{x}$, a model is considered as irreducible if any subset of an object, $\mathbf{y} \subseteq \mathbf{x}_k$ is not functionally independent of the complement of the subset contained within the object, $\mathbf{y}^c \cap \mathbf{x}$ as expressed by the Jacobin rank inequality in equation 5 in [6].

Assumption 4 (Compositionality)**.** Compositionality as defined by is a structure imposed on the slot decoder f_d which implies that each image pixel is a function of at most one slot representation, thereby enforcing a local sparsity structure on the Jacobian matrix of f_d [6].

Remark 2. The compositionality is guaranteed by explicitly minimising the compositionality contrast but was empirically observed to be implicitly satisfied in the case of additive decoders [6]. Later, [47] showed this as the property of additive decoder models. However, the additive decoders studied by [47] are not expressive enough to represent the "masked decoders" typically used in object-centric representation learning, which stems from the normalization of the alpha masks. This means some care must be taken in extrapolating the results in [47] to the models we use in practice. Additionally, additive decoders scale linearly in the number of slots K, so some less significant scalability issues remain relative to state-of-the-art non-additive decoders (e.g. using Transformers).

Assumption 5 ($\mathbf{u}$ task)**.** Conditioning latent variables on an observed variable to yield identifiable models. The main assumption is that conditioning on a (potentially observed) variable $\mathbf{u}$ renders the latent variables independent of each other [37].

Assumption 6 (Object Sufficiency)**.** A model is said to be object sufficient iff there are no additional objects in the original data distributions other than the ones expressed in training data.

Assumption 7 (Decoder Injectivity)**.** The function $f_d : \mathcal{S} \to \mathcal{X}$ mapping from slot space to image space is a non-linear piecewise affine injective function. That is, it specifies a unique one-to-one mapping between slots and images.

Remark 3. In practice, we use a monotonically increasing decoder with leakyReLU activation which should encourage injectivity behaviour [38, 37].

Assumption 8 (*Weak* Injectivity [43])**.** Let $f : \mathcal{Z} \to \mathcal{X}$ be a mapping between latent space and image space, where $\dim(\mathcal{Z}) \leq \dim(\mathcal{X})$. The mapping f_d is weakly injective if there exists $\mathbf{x}_0 \in \mathcal{X}$ and $\delta > 0$ such that $|f^{-1}(\{\mathbf{x}\})| = 1, \forall\, \mathbf{x} \in B(\mathbf{x}_0, \delta) \cap f(\mathcal{Z})$, and $\{\mathbf{x} \in \mathcal{X} :|\, f^{-1}(\{\mathbf{x}\}) \,|= \infty\} \subseteq f(\mathcal{Z})$ has measure zero w.r.t. to the Lebesgue measure on $f(\mathcal{Z})$.

Remark 4. In words, Assumption 8 says that a mapping f_d is weakly injective if: (i) in a small neighbourhood around a specific point $\mathbf{x}_0 \in \mathcal{X}$ the mapping is injective – meaning each point in this neighbourhood maps to exactly one point in the latent space $\mathcal{Z}$; and (ii) while f_d may not be globally injective, the set of points in $\mathcal{X}$ that map back to an infinite number of points in $\mathcal{Z}$ (non-injective points) is almost non-existent in terms of the Lebesgue measure on the image of $\mathcal{Z}$ under f_d.

B Aggregate Posterior Proofs

Lemma 1 (Aggregate Posterior Mixture) Given that probabilistic slot attention induces a local (per-datapoint $\mathbf{x} \in \{\mathbf{x}_i\}_{i=1}^M$) GMM with K components, the aggregate posterior $q(\mathbf{z})$ obtained by marginalizing out $\mathbf{x}$ is a non-degenerate global Gaussian mixture with MK components given by:

$$q(\mathbf{z}) = \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{K}\widehat{\boldsymbol{\pi}}_{ik}\mathcal{N}\left(\mathbf{z}; \widehat{\boldsymbol{\mu}}_{ik}, \widehat{\boldsymbol{\sigma}}_{ik}^2\right). \tag{13}$$

Proof. We begin by noting that the aggregate posterior $q(\mathbf{z})$ is the optimal prior $p(\mathbf{z})$ so long as our posterior approximation $q(\mathbf{z} \mid \mathbf{x})$ is close enough to the true posterior $p(\mathbf{z} \mid \mathbf{x})$, since for a dataset $\{\mathbf{x}_i\}_{i=1}^M$ we have that:

$$\begin{aligned}
p(\mathbf{z}) &= \int p(\mathbf{z} \mid \mathbf{x})p(\mathbf{x})d\mathbf{x} && (14)\\
&= \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[p(\mathbf{z} \mid \mathbf{x})\right] && (15)\\
&\approx \frac{1}{M}\sum_{i=1}^{M} p(\mathbf{z} \mid \mathbf{x}_i) && (16)\\
&\approx \frac{1}{M}\sum_{i=1}^{M} q(\mathbf{z} \mid \mathbf{x}_i) && (17)\\
&=: q(\mathbf{z}), && (18)
\end{aligned}$$

where we approximated $p(\mathbf{x})$ using the empirical distribution, then substituted in the approximate posterior and marginalized out $\mathbf{x}$. This observation was first made by [28] and we use it to motivate our setup.

In our case, probabilistic slot attention (Algorithm 1) fits a (local) GMM to each latent variable sampled from the approximate posterior: $\mathbf{z} \sim q(\mathbf{z} \mid \mathbf{x}_i)$, for $i = 1, \ldots, M$. Let $f(\mathbf{z})$ denote the (local) GMM density, its expectation is given by:

$$\begin{aligned}
\mathbb{E}_{p(\mathbf{x}),q(\mathbf{z}|\mathbf{x})}\left[f(\mathbf{z})\right] &= \iint p(\mathbf{x})q(\mathbf{z} \mid \mathbf{x})f(\mathbf{z})d\mathbf{x}d\mathbf{z} && (19)\\
&\approx \iint \frac{1}{M}\sum_{i=1}^{M}\delta(\mathbf{x}-\mathbf{x}_i)q(\mathbf{z} \mid \mathbf{x})f(\mathbf{z})d\mathbf{x}d\mathbf{z} && (20)\\
&= \int \frac{1}{M}\sum_{i=1}^{M} q(\mathbf{z} \mid \mathbf{x}_i)f(\mathbf{z})d\mathbf{z} && (21)\\
&= \int \frac{1}{M}\sum_{i=1}^{M}\mathcal{N}\big(\mathbf{z}; \boldsymbol{\mu}(\mathbf{x}_i), \boldsymbol{\sigma}^2(\mathbf{x}_i)\big)\cdot\sum_{k=1}^{K}\boldsymbol{\pi}_{ik}\mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_{ik}, \boldsymbol{\sigma}_{ik}^2\right)d\mathbf{z} && (22)\\
&\approx \int \frac{1}{M}\sum_{i=1}^{M}\delta(\mathbf{z}-\boldsymbol{\mu}(\mathbf{x}_i))\cdot\sum_{k=1}^{K}\boldsymbol{\pi}_{ik}\mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_{ik}, \boldsymbol{\sigma}_{ik}^2\right)d\mathbf{z} && (23)\\
&= \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{K}\boldsymbol{\pi}_{ik}\mathcal{N}\left(\boldsymbol{\mu}(\mathbf{x}_i); \boldsymbol{\mu}_{ik}, \boldsymbol{\sigma}_{ik}^2\right) && (24)\\
&=: q(\mathbf{z}), && (25)
\end{aligned}$$

where we again used the empirical distribution approximation of $p(\mathbf{x})$, and the following basic identity of the Dirac delta to simplify: $\int \delta(\mathbf{x}-\mathbf{x}')f(\mathbf{x})d\mathbf{x} = f(\mathbf{x}')$.

For the general case, however, we must instead compute the product of $q(\mathbf{z} \mid \mathbf{x})$ and $f(\mathbf{z})$ rather than use a Dirac delta approximation as in Equation 23. To that end we may proceed as follows w.r.t. to

each datapoint $\mathbf{x}_i$:

$$q(\mathbf{z} \mid \mathbf{x}_i) \cdot f(\mathbf{z}) = \mathcal{N}\big(\mathbf{z}; \boldsymbol{\mu}(\mathbf{x}_i), \boldsymbol{\sigma}^2(\mathbf{x}_i)\big) \cdot \sum_{k=1}^{K} \boldsymbol{\pi}_{ik} \mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_{ik}, \boldsymbol{\sigma}_{ik}^2\right) \tag{26}$$

$$= \sum_{k=1}^{K} \boldsymbol{\pi}_{ik} \left[\mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_{ik}, \boldsymbol{\sigma}_{ik}^2\right) \cdot \mathcal{N}\big(\mathbf{z}; \boldsymbol{\mu}(\mathbf{x}_i), \boldsymbol{\sigma}^2(\mathbf{x}_i)\big)\right] \tag{27}$$

$$= \sum_{k=1}^{K} \widehat{\boldsymbol{\pi}}_{ik} \mathcal{N}\left(\mathbf{z}; \widehat{\boldsymbol{\mu}}_{ik}, \widehat{\boldsymbol{\sigma}}_{ik}^2\right), \tag{28}$$

where the posterior parameters of the resulting mixture are given in closed-form by:

$$\widehat{\boldsymbol{\sigma}}_{ik}^2 = \left(\frac{1}{\boldsymbol{\sigma}_{ik}^2} + \frac{1}{\boldsymbol{\sigma}^2(\mathbf{x}_i)}\right)^{-1}, \qquad \widehat{\boldsymbol{\mu}}_{ik} = \widehat{\boldsymbol{\sigma}}_{ik}^2 \left(\frac{\boldsymbol{\mu}(\mathbf{x}_i)}{\boldsymbol{\sigma}^2(\mathbf{x}_i)} + \frac{\boldsymbol{\mu}_{ik}}{\boldsymbol{\sigma}_{ik}^2}\right), \tag{29}$$

which are the standard distributional parameters obtained from a product of two Gaussians.

For the updated mixture coefficients $\widehat{\boldsymbol{\pi}}_{ik}$, we propose a principled way to include a posterior-weighted contribution of each mode to the new mixture coefficients. First, note that $(\boldsymbol{\pi}_{i1}, \boldsymbol{\pi}_{i2}, \ldots, \boldsymbol{\pi}_{iK})$ are parameters of a multinomial distribution as $\sum_{k=1}^{K} \boldsymbol{\pi}_{ik} = 1$, for each datapoint $\mathbf{x}_i$. Since the Dirichlet distribution is the conjugate prior of the multinomial distribution, we can place a Dirichlet prior over the mixing coefficients for each datapoint, then update it to a posterior using the data. Concretely, we place a symmetric Dirichlet prior over the mixing coefficients as follows:

$$(\boldsymbol{\pi}_{i1}, \boldsymbol{\pi}_{i2}, \ldots, \boldsymbol{\pi}_{iK}) \sim \text{Dirichlet}\left(\boldsymbol{\alpha}_{i1}, \boldsymbol{\alpha}_{i2}, \ldots, \boldsymbol{\alpha}_{iK}\right), \quad \text{for} \quad i = 1, 2, \ldots, M, \tag{30}$$

where $\boldsymbol{\alpha}_i \in \mathbb{R}^K$ are the concentration parameters of the i^{th} Dirichlet distribution, and $\forall i, k: \ \boldsymbol{\alpha}_{ik} = 1$, indicating uniformity over the open standard $(K-1)$-simplex. To compute the posterior Dirichlet distribution we calculate 'pseudo-counts' by integrating the product of the posterior density $q(\mathbf{z} \mid \mathbf{x}_i)$ with each one of the K modes of the Gaussian mixture, thereby measuring a posterior-weighted contribution of each mode k to the new aggregate mixture:

$$c_{ik} = \int \mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_{ik}, \boldsymbol{\sigma}_{ik}^2\right) \cdot \mathcal{N}\big(\mathbf{z}; \boldsymbol{\mu}(\mathbf{x}_i), \boldsymbol{\sigma}^2(\mathbf{x}_i)\big) d\mathbf{z}, \qquad \text{for} \quad i = 1, 2, \ldots, M, \tag{31}$$

which we can then use as pseudo-counts to compute the Dirichlet posterior:

$$(\widehat{\boldsymbol{\pi}}_{i1}, \widehat{\boldsymbol{\pi}}_{i2}, \ldots, \widehat{\boldsymbol{\pi}}_{iK}) \mid (c_{i1}, c_{i2}, \ldots, c_{iK}) \sim \text{Dirichlet}\left(\boldsymbol{\alpha}_{i1} + c_{i1}, \boldsymbol{\alpha}_{i2} + c_{i2}, \ldots, \boldsymbol{\alpha}_{iK} + c_{iK}\right), \tag{32}$$

for $i = 1, 2, \ldots, M$. Each posterior probability is then readily given by the mean estimate

$$\widehat{\boldsymbol{\pi}}_{ik} = \frac{\boldsymbol{\alpha}_{ik} + c_{ik}}{\sum_{j=1}^{K} (\boldsymbol{\alpha}_{ij} + c_{ij})} \implies \sum_{k=1}^{K} \widehat{\boldsymbol{\pi}}_{ik} = 1. \tag{33}$$

Putting everything together, the aggregated posterior is therefore given by:

$$q(\mathbf{z}) = \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} \widehat{\boldsymbol{\pi}}_{ik} \mathcal{N}\left(\mathbf{z}; \widehat{\boldsymbol{\mu}}_{ik}, \widehat{\boldsymbol{\sigma}}_{ik}^2\right), \qquad \text{where} \qquad \mathbf{z} \sim q(\mathbf{z} \mid \mathbf{x}), \tag{34}$$

which concludes the proof. □

Corollary 4. *The aggregate posterior $q(\mathbf{z})$ is a non-degenerate Gaussian mixture, in the sense that it is a well-defined probability distribution, as the updated mixture coefficients sum to 1 over the number of components $M \times K$.*

Proof. Recall from Lemma 1 that the aggregate posterior $q(\mathbf{z})$ – obtained by marginizaling out $\mathbf{x}$ from a probabilistic slot attention model – is a mixture distribution of $M \times K$ components with the following parameters:

$$\left\{\widehat{\boldsymbol{\pi}}_{ik}, \widehat{\boldsymbol{\mu}}_{ik}, \widehat{\boldsymbol{\sigma}}_{ik}^2\right\}, \qquad \text{for} \quad i = 1, 2, \ldots, M, \quad \text{and} \quad k = 1, 2, \ldots, K. \tag{35}$$

To verify that $q(\mathbf{z})$ is a non-degenerate mixture, we observe the following implication:

$$\sum_{k=1}^{K} \widehat{\boldsymbol{\pi}}_{ik} = 1, \qquad \text{for} \quad i = 1, 2, \ldots, M, \tag{36}$$

due to the Dirichlet posterior update in Equation 33, and therefore

$$\implies \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} \widehat{\boldsymbol{\pi}}_{ik} = \frac{1}{M} \sum_{i=1}^{M} 1 = \frac{1}{M} \cdot M = 1 \tag{37}$$

$$\implies \sum_{i=1}^{M} \sum_{k=1}^{K} \frac{\widehat{\boldsymbol{\pi}}_{ik}}{M} = 1, \tag{38}$$

which says that the scaled sum of the mixing proportions of all K components in all M GMMs must equal 1, proving that the associated aggregate posterior mixture $q(\mathbf{z})$ is a well-defined probability distribution. □

Theorem 1 (Mixture Distribution of Concatenated Slots)**.** Let f_s denote a permutation equivariant probabilistic slot attention function such that $f_s(\mathbf{z}, P\mathbf{s}^t) = Pf_s(\mathbf{z}, \mathbf{s}^t)$, where $P \in \{0, 1\}^{K \times K}$ is an arbitrary permutation matrix. Let $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K) \in \mathbb{R}^{Kd}$ be a random variable defined as the concatenation of K individual sampled slots, where each slot is Gaussian distributed within a K-component mixture: $\mathbf{s}_k \sim \mathcal{N}(\mathbf{s}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \in \mathbb{R}^d, \forall\, k \in \{1, \ldots K\}$. Then, it holds that $\mathbf{s}$ is also Gaussian mixture distributed comprising $K!$ mixture components:

$$p(\mathbf{s}) = \sum_{p=1}^{K!} \boldsymbol{\pi}_p \mathcal{N}(\mathbf{s}; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p), \quad \text{where} \quad \boldsymbol{\pi} \in \Delta^{K!-1}, \quad \boldsymbol{\mu}_p \in \mathbb{R}^{Kd}, \quad \boldsymbol{\Sigma}_p \in \mathbb{R}^{Kd \times Kd}. \tag{39}$$

Proof. Each slot $\mathbf{s}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is sampled independently from $\mathbf{s}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, for any $j \neq k$, meaning they are conditionally independent given the latent mixture component assignment. Thus, the concatenated slots variable $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K)$, can be described by a Kd-dimensional multivariate Gaussian distribution with a block diagonal covariance structure as follows:

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_{\pi(1)} \\ \mathbf{s}_{\pi(2)} \\ \vdots \\ \mathbf{s}_{\pi(K)} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_{\pi(1)} \\ \boldsymbol{\mu}_{\pi(2)} \\ \vdots \\ \boldsymbol{\mu}_{\pi(K)} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\pi(1)} & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Sigma}_{\pi(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{\Sigma}_{\pi(K)} \end{bmatrix} \right), \tag{40}$$

where $\pi : [K] \to [K]$ is a permutation function of the set $[K] \coloneqq \{1, 2, \ldots, K\}$. Since the slot attention function f_s is permutation equivariant, there exist $K!$ possible ways to concatenate K slots, and each permutation induces a mode within a Gaussian mixture living in $\mathbb{R}^{Kd}$ space. Since each permutation of the slots is equally likely, the mixture coefficients are given by:

$$\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_{K!}), \qquad \text{where} \qquad \boldsymbol{\pi}_p = \frac{1}{K!} \qquad \forall p \in \{1, 2, \ldots, K!\} \tag{41}$$

$$\implies \sum_{p=1}^{K!} \boldsymbol{\pi}_p = 1, \tag{42}$$

which concludes the proof. □

Remark 5. Based on the above result, it is evident that concatenating $K \geq 2$ unique slots can be viewed as a sample from a GMM with $K!$ components. Constructing a scene requires *at least* two unique slots, one for the background and one for an object, thus supporting our theory regarding slot composition.

C Injective Decoders: Sufficient Conditions

In this section, we provide a theoretical overview of the decoder architecture we use and offer sufficient conditions for a leaky-ReLU decoder to be *weakly* injective in the sense of Assumption 8 as shown and adapted from [43].

Definition 4 (Piece-wise Decoders, [43])**.** Let $\mathbf{s} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_K\}$ denote a given set of sampled z_i in each mixture component (K slots) in the GMM, $P(\mathcal{Z})$. Let $\sigma : \mathbb{R} \to \mathbb{R}$ denote the leaky-ReLU activation function, and let $m = n_0, n_1, n_2, \dots, n_t = n$ and $H(n_1, n_2)$ denote the set of full-rank affine functions $h_i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_j}$. We consider piece-wise functions mapping each slot representation $\mathbf{s} \in \mathcal{S} \in \mathbb{R}^m \times K$ to an image $\mathbf{x} \in \mathcal{X} \in \mathbb{R}^n$ in the output space, $\mathcal{F}_\sigma^{mK \to n} : \mathcal{S} \to \mathcal{X}$, of the form below:

$$\mathcal{F}_\sigma^{n_0,\dots,n_t} = \Big\{ h_t \circ \sigma \circ h_{t-1} \circ \sigma \circ \cdots \sigma \circ h_1 \mid h_i \in H(n_{i-1}, n_i) \Big\}. \tag{43}$$

The following Corollary, corollary and proofs are adapted from [43].

Corollary 5. *Given $f_d \in \mathcal{F}_\sigma^{m \hookrightarrow n}$ where $f_d = h_t \circ \sigma \circ h_{t-1} \circ \sigma \circ \dots \sigma \circ h_1$, f_d is injective.*

Proof. Each affine function h_i has full column rank and is therefore both injective and invertible. Since the activation function σ is also injective, we get that f_d is injective and invertible. □

Corollary 6. *Let $f_d = h_t \circ \sigma \circ h_{t-1} \circ \sigma \circ \dots \sigma \circ h_1 \in \mathcal{F}_\sigma^{m \to n}$ where $m = n_0 \leq n_1 \leq \cdots \leq n_k = n$. Given h_i is affine and invertible, then for almost all $x \in f_d(\mathbb{R}^m)$ there exists δ such that f_d^{-1} is a well-defined affine function, h_i on $B(x, \delta) \cap f_d(\mathbb{R}^m)$.*

Proof. We know f_d is a piecewise affine function which is invertible. Therefore, it simply follows, $\forall y \in B(x, \delta)$ there exists δ such that f_d^{-1} is an affine function in the domain $B(x, \delta)$. One can therefore deduce there exists some affine function h_i where $f_d^{-1} = h_i^{-1}$ in the domain $B(x, \delta)$. □

D Slot Identifiability Proof

Definition 5 (Slot Identifiability [6])**.** Given a diffeomorphic ground truth function $f : \mathcal{S} \to \mathcal{X}$, and inference model $\hat{g} : \mathcal{X} \to \mathcal{S}$, $\hat{g}$ correctly slot identifies every object $\mathbf{x}_j \in \mathcal{X}$ with the ground slot $\mathbf{s}_k \in \mathcal{S}$ via $\hat{\mathbf{s}}_k = \hat{g}(f(\mathbf{s}_k))$ if there exists a unique slot $k \in [K]$ for all $\mathbf{x}_j \in \mathcal{X}$, and there exists an invertible diffeomorphism, or in our case an affine transformation, such that $\mathbf{s}_k = h(\hat{\mathbf{s}}_k)$ for all $\mathbf{s}_k \in \mathcal{S}$.

Remark 6. [6] established that *compositionality* (Assumption 4) and *irreducibility* (Assumption 3) of the decoder f_d is required for slot identifiability in the sense of Definition 5.

Summary & Intuition. The following theorems and proof extend the identifiability results of [43] to slot-attention models, we include all the proofs and details for the sake of completion. In this work, we do not consider the irreducibility criteria in [6] and define slot identifiability only by an injective mapping of each slot to subspaces representing objects in a scene to satisfy compositionality without the use of computationally heavy methods such as additive decoders and compositional contrast. We show identifiability of each slot representation up to affine transformation by passing a concatenation of samples from each mixture component which represents slots through a piecewise function in Definition 4 representing the decoder, f_d. The trick in this proof lies in our observation of the fact that a concatenation of samples from each slot mixture component is a sample from a high dimensional GMM $p(\mathbf{s})$, as highlighted in Theorem 1. We then use the identifiability results of [43] to show $p(\mathbf{s})$ is identifiable up to affine transformation. This then implies identifiability up to affine transformation of the aggregate posterior $q(\mathbf{z})$, a non-degenerate GMM by Lemma 1 where a sample from a mixture component in $q(\mathbf{z})$ represents an individual slot representation. This contrasts with [43] which does not consider identifiability of the aggregate posterior and its mixture components in the context of slot representation learning.

In order to proceed, we begin by stating three key theorems defined and proven in the work of [43] which are essential for our slot identifiability proof. First, we restate the definition of a *generic point* as outlined by [43] below.

Definition 6. A point $\mathbf{x} \in f_d(\mathbb{R}^m) \subseteq \mathbb{R}^n$ is generic if there exists $\delta > 0$, such that $f_d : B(\mathbf{s}, \delta) \to \mathbb{R}^n$ is affine for every $\mathbf{s} \in f_d^{-1}(\{\mathbf{x}\})$

Theorem 7 (Kivva et al. [43]). *Given $f_d : \mathbb{R}^m \to \mathbb{R}^n$ is a piecewise affine function such that $\{\mathbf{x} \in \mathbb{R}^n : |f_d^{-1}(\{\mathbf{x}\})| = \infty\} \subseteq f_d(\mathbb{R}^m)$ has measure zero with respect to the Lebesgue measure on $f_d(\mathbb{R}^m)$, this implies $\dim(f_d(\mathbb{R}^m)) = m$ and almost every point in $f_d(\mathbb{R}^m)$ (with respect to the Lebesgue measure on $f_d(\mathbb{R}^m)$) is generic with respect to f_d.*

Theorem 8 (Kivva et al. [43]). *Consider a pair of finite GMMs in $\mathbb{R}^m$:*

$$\mathbf{y} = \sum_{j=1}^{J} \boldsymbol{\pi}_j \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), \qquad \text{and} \qquad \mathbf{y}' = \sum_{j=1}^{J'} \boldsymbol{\pi}'_j \mathcal{N}(\mathbf{y}'; \boldsymbol{\mu}'_j, \boldsymbol{\Sigma}'_j). \tag{44}$$

Assume that there exists a ball $B(\mathbf{x}, \delta)$ such that $\mathbf{y}$ and $\mathbf{y}'$ induce the same measure on $B(\mathbf{x}, \delta)$. Then $\mathbf{y} \equiv \mathbf{y}'$, and for some permutation τ we have that $\boldsymbol{\pi}_i = \boldsymbol{\pi}'_{\tau(i)}$ and $(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = (\boldsymbol{\mu}'_{\tau(i)}, \boldsymbol{\Sigma}'_{\tau(i)})$.

Theorem 9 (Kivva et al. [43]). *Given $\mathbf{z} \sim \sum_{i=1}^{J} \boldsymbol{\pi}_i \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ and $\mathbf{z}' \sim \sum_{j=1}^{J'} \boldsymbol{\pi}'_j \mathcal{N}(\mathbf{z}'; \boldsymbol{\mu}'_j, \boldsymbol{\Sigma}'_j)$ and $f_d(\mathbf{z})$ and $\tilde{f}_d(\mathbf{z}')$ are equally distributed. We can assume for $\mathbf{x} \in \mathbb{R}^n$ and $\delta > 0$, f_d is invertible on $B(\mathbf{x}, 2\delta) \cap f_d(\mathbb{R}^m)$. This implies that there exists $\mathbf{x}_1 \in B(\mathbf{x}, \delta)$ and $\delta_1 > 0$ such that both f_d and $\tilde{f}_d$ are invertible on $B(\mathbf{x}_1, \delta_1) \cap f_d(\mathbb{R}^m)$.*

We next propose our slot identifiability result below.

Theorem 2 ($\sim_s$-Identifiable Slot Representations)**.** Given that the aggregate posterior $q(\mathbf{z})$ is an optimal, non-degenerate mixture prior over slot space (Lemma 1), $f_d : \mathcal{S} \to \mathcal{X}$ is a piecewise affine weakly injective mixing function (Assumption 8), and the slot representation, $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_K)$ can be observed as a sample from a GMM (Theorem 1), then $p(\mathbf{s})$ is identifiable as per Definition 3.

Proof. The proof extends from [43] to slot-based models. Given two piece-wise affine functions $f_d, \tilde{f}_d : \mathcal{S} \to \mathcal{X}, \forall k \in [K]$, let $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_K), \ni \mathbf{s}_k \sim \mathcal{N}(\mathbf{s}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ and $\mathbf{s}' = (\mathbf{s}'_1, \ldots, \mathbf{s}'_K), \ni \mathbf{s}'_k \sim \mathcal{N}(\mathbf{s}'_k; \boldsymbol{\mu}'_k, \boldsymbol{\Sigma}'_k)$ be a pair of slot representations constructed by sampling and concatenating each mixture component (i.e. slots) from two distinct GMMs. As proven in Theorem 1, given individual sampled slots $\mathbf{s}_k$ are conditionally independent given the mixture component k, then a concatenated sample is from a higher dimensional GMM in $\mathbb{R}^{Kd}$. Now, suppose for the sake of argument that $f_d(\mathcal{S})$ and $\tilde{f}_d(\mathcal{S}')$ are equally distributed. We assume that there exists $\mathbf{x} \in \mathcal{X}$ and $\delta > 0$ such that f_d and $\tilde{f}_d$ are invertible and piecewise affine on $B(\mathbf{x}, \delta) \cap f_d(\mathcal{S})$, which implies $\dim f_d(\mathcal{S}) = |\mathcal{S}|$.

We now restrict the space $B(\mathbf{x}, \delta)$ to a subspace $B(\mathbf{x}', \delta')$ where $\mathbf{x} \in B(\mathbf{x}', \delta')$ such that f_d and $\tilde{f}_d$ are now invertible and affine on $B(\mathbf{x}', \delta') \cap f_d(\mathcal{S})$. Next, we let $L \subseteq \mathcal{X}$ be an $|\mathcal{S}|$-dimensional affine subspace (assuming $|\mathcal{X}| \geq |\mathcal{S}|$), such that $B(\mathbf{x}', \delta') \cap f_d(\mathcal{S}) = B(\mathbf{x}', \delta') \cap L$. We also define $h_f, h_{\tilde{f}} : \mathcal{S} \to L$ to be a pair of invertible affine functions where $h_f^{-1}(B(\mathbf{x}', \delta') \cap L) = f_d^{-1}(B(\mathbf{x}', \delta') \cap L)$ and $h_{\tilde{f}}^{-1}(B(\mathbf{x}', \delta') \cap L) = \tilde{f}_d^{-1}(B(\mathbf{x}', \delta') \cap L)$.

Therefore, this implies $h_f(\mathbf{s})$ and $h_{\tilde{f}}(\mathbf{s}')$ are finite GMMs which coincide on $B(\mathbf{x}', \delta') \cap L$ and $h_f(\mathbf{s}) \equiv h_{\tilde{f}}(\mathbf{s}')$ based on Theorem 8. Given, $h = h_{\tilde{f}}^{-1} \circ h_f$ and $h_f(\mathbf{s})$ and $h_{\tilde{f}}(\mathbf{s}')$ then h is an affine transformation such that $h(\mathbf{s}) = \mathbf{s}'$.

Given Theorems 7 and 9, there exists a point $\mathbf{x} \in f_d(\mathcal{S})$ that is generic with respect f_d and $\tilde{f}_d$ and invertible on $B(\mathbf{x}, \delta) \cap f_d(\mathcal{S})$. Having established that there is an affine transformation $h(\mathbf{s}) = \mathbf{s}'$ and two invertible piecewise affine functions f_d and $\tilde{f}_d$ on $B(\mathbf{x}, \delta) \cap f_d(\mathcal{S})$, this implies that $p(\mathbf{s})$ is identifiable up to an affine transformation and permutation of $\mathbf{s}_k \in \mathbf{s}$, which concludes the proof. □

Corollary 3 (Individual Slot Identifiability)**.** If the concatenated slot distribution $p(\mathbf{s})$ is $\sim_s$-identifiable (Theorem 2) then this implies $q(\mathbf{z})$ is identifiable up to affine transformation and permutation of the slots, $\mathbf{s}_k$ and therefore each slot distribution $\mathbf{s}_k \sim \mathcal{N}(\mathbf{s}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \in \mathbb{R}^d, \forall\, k \in \{1, \ldots K\}$ is also identifiable up to an affine transformation.

Proof. Given Theorem 8, we know that each higher dimensional mixture component in $p(\mathbf{s})$ induces the same measure on $B(\mathbf{x}, \delta)$ and hence for some permutation τ we have that $(\boldsymbol{\mu}_{\pi(i)}, \boldsymbol{\Sigma}_{\pi(i)}) =$

Algorithm 2 Probabilistic Slot Attention (no $\mathbf{k}$ or $\mathbf{v}$)

Input: $\mathbf{z} = f_e(\mathbf{x}) \in \mathbb{R}^{N \times d}$ ▷ input representation

$\forall k, \boldsymbol{\pi}(0)_k \leftarrow 1/K, \boldsymbol{\mu}(0)_k \sim \mathcal{N}(0, \mathbf{I}_d), \boldsymbol{\sigma}(0)_k^2 \leftarrow \mathbf{1}_d$

for $t = 0, \ldots, T-1$

$$A_{nk} \leftarrow \frac{\boldsymbol{\pi}(t)_k \mathcal{N}\left(\mathbf{z}_n; \boldsymbol{W}_q \boldsymbol{\mu}(t)_k, \boldsymbol{\sigma}(t)_k^2\right)}{\sum_{j=1}^{K} \boldsymbol{\pi}(t)_j \mathcal{N}\left(\mathbf{z}_n; \boldsymbol{W}_q \boldsymbol{\mu}(t)_j, \boldsymbol{\sigma}(t)_j^2\right)}$$

$\hat{A}_{nk} \leftarrow A_{nk} / \sum_{l=1}^{N} A_{lk}$ ▷ normalize attention

$\boldsymbol{\mu}(t+1)_k \leftarrow \sum_{n=1}^{N} \hat{A}_{nk} \mathbf{z}_n$ ▷ update slot mean

$\boldsymbol{\sigma}(t+1)_k^2 \leftarrow \sum_{n=1}^{N} \hat{A}_{nk} \left(\mathbf{z}_n - \boldsymbol{\mu}(t+1)_k\right)^2$

$\boldsymbol{\pi}(t+1)_k \leftarrow \sum_{n=1}^{N} A_{nk} / N$ ▷ update mixing coeff.

return $\boldsymbol{\mu}(T), \boldsymbol{\sigma}(T)^2$ ▷ K slot distributions

Algorithm 3 Probabilistic Slot Attention V.2 (PSA-PROJ)

Input: $\mathbf{z} = f_e(\mathbf{x}) \in \mathbb{R}^{N \times d}$ ▷ input representation

$\mathbf{k} \leftarrow \boldsymbol{W}_k \mathbf{z} \in \mathbb{R}^{N \times d}$ ▷ compute keys

$\forall k, \boldsymbol{\pi}(0)_k \leftarrow 1/K, \boldsymbol{\mu}(0)_k \sim \mathcal{N}(0, \mathbf{I}_d), \boldsymbol{\sigma}(0)_k^2 \leftarrow \mathbf{1}_d$

for $t = 0, \ldots, T-1$

$$A_{nk} \leftarrow \frac{\boldsymbol{\pi}(t)_k \mathcal{N}\left(\mathbf{k}_n; \boldsymbol{W}_q \boldsymbol{\mu}(t)_k, \boldsymbol{\sigma}(t)_k^2\right)}{\sum_{j=1}^{K} \boldsymbol{\pi}(t)_j \mathcal{N}\left(\mathbf{k}_n; \boldsymbol{W}_q \boldsymbol{\mu}(t)_j, \boldsymbol{\sigma}(t)_j^2\right)}$$

$\hat{A}_{nk} \leftarrow A_{nk} / \sum_{l=1}^{N} A_{lk}$ ▷ normalize attention

$\boldsymbol{\mu}(t+1)_k \leftarrow \sum_{n=1}^{N} \hat{A}_{nk} \boldsymbol{W}_v \mathbf{z}_n$ ▷ update slot mean

$\boldsymbol{\sigma}(t+1)_k^2 \leftarrow \sum_{n=1}^{N} \hat{A}_{nk} \left(\boldsymbol{W}_v \mathbf{z}_n - \boldsymbol{\mu}(t+1)_k\right)^2$

$\boldsymbol{\pi}(t+1)_k \leftarrow \sum_{n=1}^{N} A_{nk} / N$ ▷ update mixing coeff.

return $\boldsymbol{\mu}(T), \boldsymbol{\sigma}(T)^2$ ▷ K slot distributions

$(\boldsymbol{\mu}'_{\tau(\pi(i))}, \boldsymbol{\Sigma}'_{\tau(\pi(i))})$. Therefore, each mixture component $\mathbf{s}_{\pi(i)}$ is identifiable up to affine transformation, and permutation of slots representations in $\mathbf{s}$. Now, given sampling $\mathbf{s}_k$ is equivalent to obtaining K samples from the GMM, $q(\mathbf{z})$ and concatenating, this makes $q(\mathbf{z})$ identifiable up to affine transformation, h and permutation of slot representations in $\mathbf{s}$.

It now trivially follows that each slot representation $\mathbf{s}_k \sim \mathcal{N}(\mathbf{s}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \in \mathbb{R}^d, \forall\, k \in \{1, \ldots, K\}$ is identifiable up to affine transformation, h based on the following observed property of GMMs:

$$\sum_{k=1}^{K} \boldsymbol{\pi}_k h_\sharp \left(\mathcal{N}(\mathbf{s}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) \sim h_\sharp\Big(\sum_{k=1}^{K} \boldsymbol{\pi}_k \mathcal{N}(\mathbf{s}'_k; \boldsymbol{\mu}'_k, \boldsymbol{\Sigma}'_k)\Big), \tag{45}$$

which concludes the proof. □

E Algorithm

As a result of projection with separate query matrix is considered, the mean and variance updates are coupled with some non-linear affine transformation; for EM to have an exact solution, this transformation needs to be the identity (this can be seen when we derive the update equations for mean and variance). The resulting algorithms for these two cases are illustrated in algorithms 2 and 3.

Table 4: Quality of compositional image generation with FID measure

METHOD	CLEVR			OBJECTS-ROOM		
	FG-ARI ↑	CFID ↓	RFID ↓	FG-ARI ↑	CFID ↓	RFID ↓
SA	0.96 ± 0.01	-	41.81 ± 2.82	0.79 ± 0.06	-	16.49 ± 3.34
SA-NOA	0.62 ± 0.02	-	81.21 ± 8.72	0.41 ± 0.12	-	96.64 ± 21.33
PSA-NOA	0.84 ± 0.01	36.42 ± 8.53	68.02 ± 10.21	0.78 ± 0.04	54.55 ± 1.43	21.37 ± 1.03
PSA	0.85 ± 0.02	28.50 ± 4.27	52.70 ± 1.74	0.78 ± 0.02	20.49 ± 2.36	21.00 ± 1.43
PSA-PROJ	0.95 ± 0.00	28.79 ± 5.50	39.42 ± 8.29	0.81 ± 0.04	34.58 ± 4.32	16.85 ± 5.68

F Metrics

SIS: Slot identifiability score [6], mainly focus on R2 score between ground-truth and the estimated slot representations wrt to maximum R2 score from the models fit between each and every inferred slots. By design SIS requires a model fitting at every validation step to compute this relativistic measure, due to which the metric seems to vary quite a bit across runs, as observed in Figure 8.

SMCC: Mean correlation coefficient is a well studied metric in disentangled representational learning [39, 37], we extend this with additional permutational invariance on slot dimensions. SMCC measures the correlation between estimated and ground truth representations (or estimated representations across runs, in the case when ground truth representations are unavailable) once the slots are matched using Hungarian matching. To compute identifiability up to affine transformation along the representational axis and permutation in slots (check definition 3), similar to weak identifiability as per the definition in the paper and in [38], we use the MCC up to some affine mapping $\boldsymbol{A}$, which we learned by matching slot representations across runs. In summary, SMCC can be computed with the following three steps:

- Matching the order of slots using Hungarian matching;
- Affine transformation of slot representation;
- Followed by computing mean correlation.

Apart from variations described in Figure 8, we further analyse both the metrics by fixing the model and data; and by computing both the metrics for 10 times. We then considered the mean and variance in the performance, which is reflected as follows: SIS: $35.26 \pm \mathbf{6.46}$, SMCC: $77.83 \pm \mathbf{0.36}$. Here, the resulting variation is only the reflection of the metric, which clearly indicates the stability of SMCC over SIS.

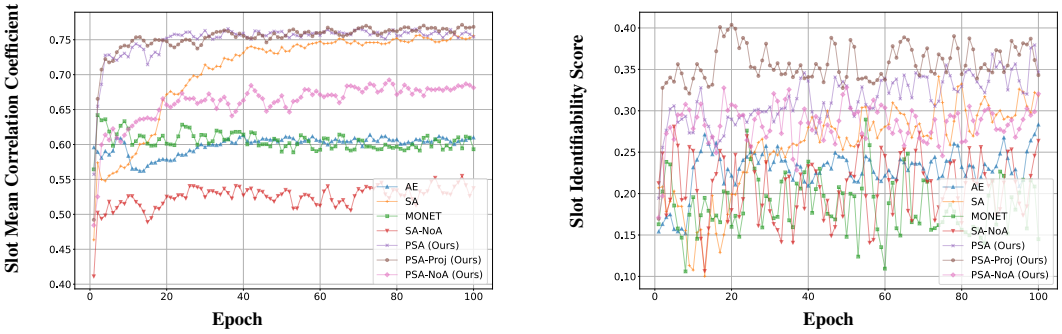


Figure 8: **Identifiability scores throughout training.** Our proposed SMCC metric (top) is much more stable than SIS (bottom) in capturing identifiability, and better in discerning differences between methods, showing substantial improvements for PSA.

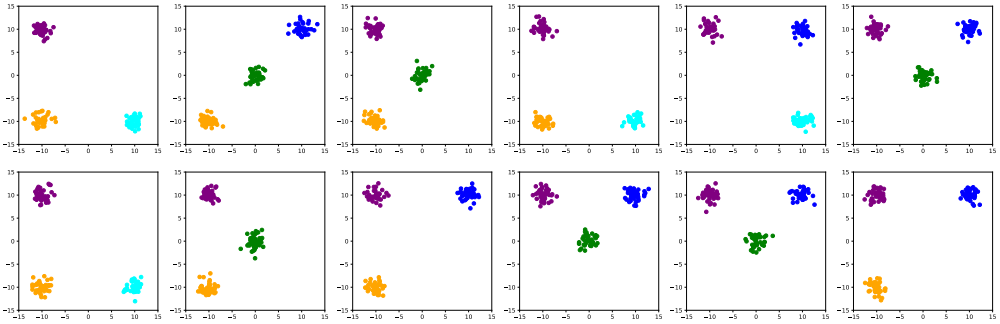


Figure 9: Random samples from the 2D synthetic dataset used in our aggregate posterior identifiability experiments. As outlined in the main text, there are in total five 'object' clusters in the dataset, and each observation contains at most three of the clusters.

G Comparison with Autoencoding Variational Bayes

As explained in Section 3, applying slot attention to a deterministic encoding $\mathbf{z} = f_e(\mathbf{x}) \in \mathbb{R}^{N \times d}$ yields a set of K object slot representations $\mathbf{s}_{1:K} := \mathbf{s}_1, \ldots, \mathbf{s}_K$. In combination, this process induces a stochastic encoder $q(\mathbf{s}_{1:K} \mid \mathbf{x})$, where the stochasticity comes from the random initialization of the slots in the first iteration: $\mathbf{s}_{1:K}^{t=0} \sim \mathcal{N}(\mathbf{s}_{1:K}; 0, \mathbf{I}) \in \mathbb{R}^{K \times d}$. Since each slot is a deterministic function of its previous state $\mathbf{s}^t := f_s(\mathbf{z}, \mathbf{s}^{t-1})$ it is possible to randomly sample initial states $\mathbf{s}^0$ and obtain stochastic estimates of the slots. However, since each transition depends on $\mathbf{z}$, which in turn depends on the input $\mathbf{x}$, *we do not get a generative model we can tractably sample from.*

This can be remedied by placing a tractable prior over $\mathbf{z}$ and using the VAE framework along the lines of [77]. Specifically, Wang et al. [77] propose the Slot-VAE, which is a generative model that integrates slot attention and the VAE framework under a two-layer hierarchical latent model. However, under their formulation, there is a key challenge in calculating the KL term as the slot attention function is permutation equivariant meaning the slots have no fixed order across the posterior and prior. To compensate for this, the authors introduce a heuristic auxiliary loss and a parallel image processing path with an additional slot attention operation which is computationally costly.

In contrast, our approach does not suffer from such drawbacks. This was achieved by designing the slot attention operation itself as probabilistic, and proving that having a local (per-datapoint) GMM results in the *aggregate slot posterior* and the concatenated slots being GMM distributed (Section 4). We then proved a new slot identifiability result using this insight (Section 5). Our use of the aggregate posterior is inspired by but differs substantially in both method and application from previous works on VAEs [28, 71, 72]. Our primary goal is not to learn a better generative prior but to obtain slot identifiable representations. Lastly, since the concatenated slots are provably GMM distributed, our model is reminiscent of a GMM-VAE [11], but with a unique slot-based structure.

H Automatic Relevance Determination of Slots

For evaluation, we calculate the MAE between the estimated and ground truth numbers of slots and measure the reduction in FLOPs achieved using the proposed ARD method. We observe MAE values of 1.03 and 0.58, and significant savings of up to **41%** and **62%** in FLOPs on the CLEVR and Objects-Room datasets respectively, when compared to using a fixed number of slots K.

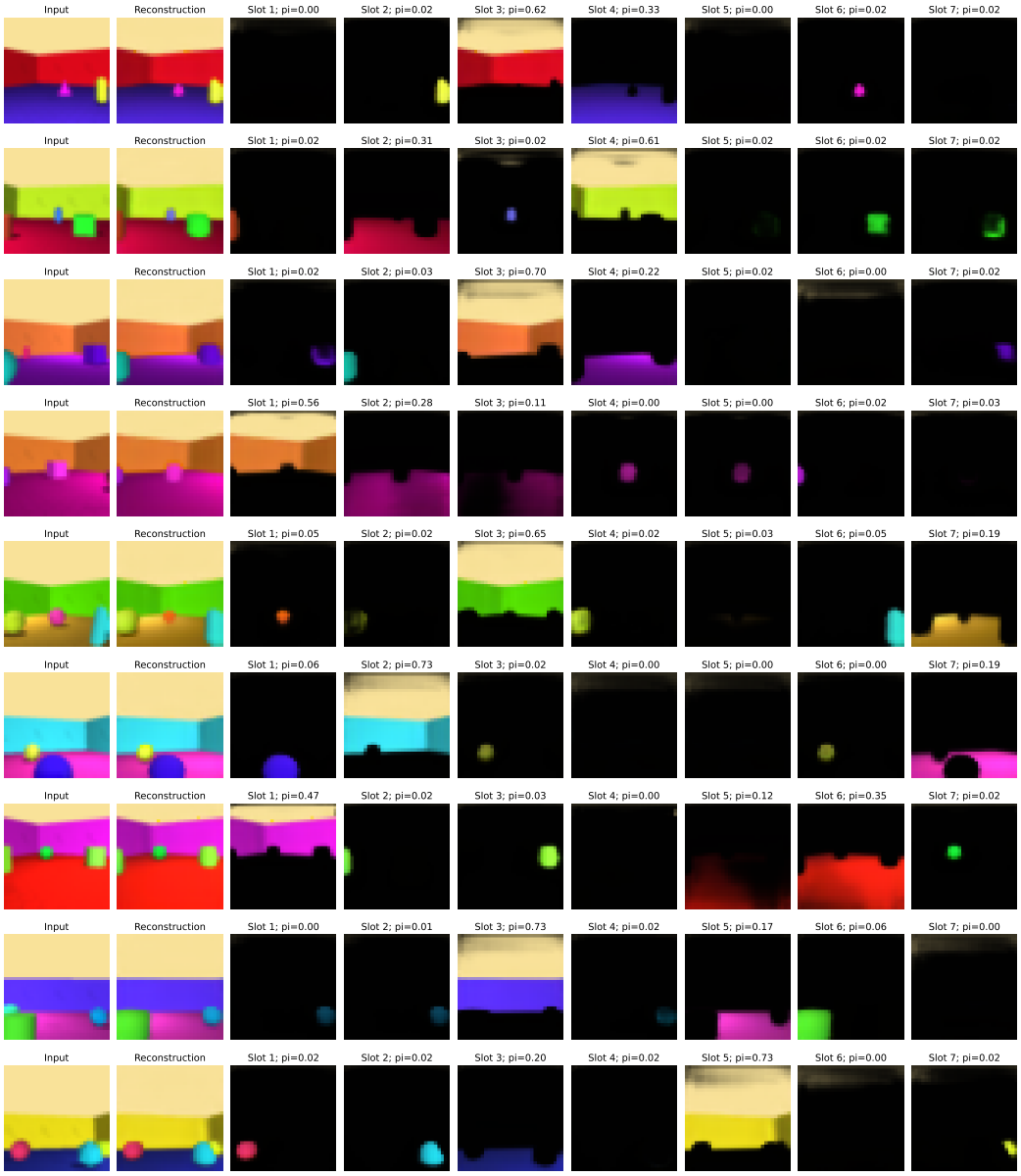


Figure 10: Automatic Relevance Determination (ARD) of slots on the OBJECTSROOM dataset. As shown, when using our proposed probabilistic slot attention (Algorithm 1), the mixing coefficients $\boldsymbol{\pi}_i \in \mathbb{R}^K, \forall i$ automatically approach zero when slots are inactive.

I Compositionality Results

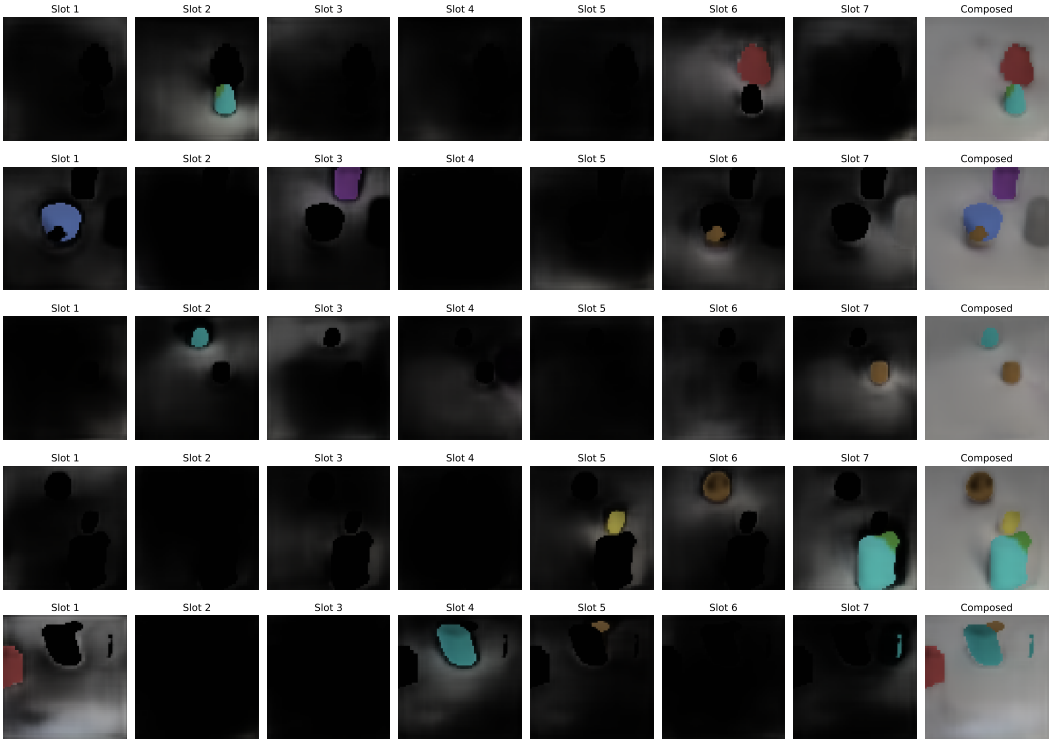


Figure 11: Aggregate posterior sampling for image composition on CLEVR dataset.

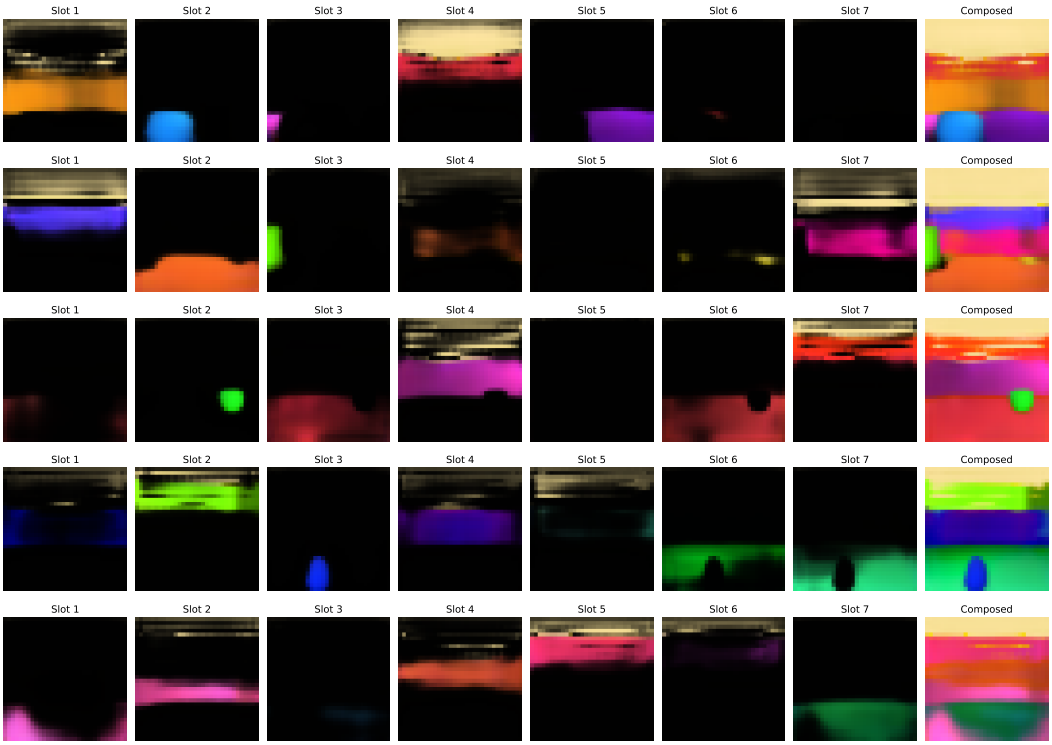


Figure 12: Aggregate posterior sampling for image composition on OBJECTSROOM dataset.

Figure 13: Visualizations of attention and alpha masks on the Pascal VOC2012 dataset are shown, with alpha masks on the left and attention masks on the right, for a PSA-MLP model using a DINO feature extractor. In the figure above, the images from left to right represent: the original image, ground-truth segmentation, alpha mask segmentation, individual entities grouped in the alpha mask, slot attention segmentation mask, and individual entities grouped in the slot attention mask.

Figure 14: Visualizations of attention and alpha masks on the Pascal VOC2012 dataset are shown, with alpha masks on the left and attention masks on the right, for a PSA-Transformer model using a DINO feature extractor. In the figure above, the images from left to right represent: the original image, ground-truth segmentation, alpha mask segmentation, individual entities grouped in the alpha mask, slot attention segmentation mask, and individual entities grouped in the slot attention mask.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly state the limitations in existing literature and our proposed analysis to overcome these limitation, which we further back with theoretical guarantees and experimental observations.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a detailed discussion on limitations and explicitly list all the assumptions made, which serves nicely for future works.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide rigorous mathematical proof all our theoretical claims.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We clearly list all the set of hyperparameters used in the analysis.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide codebase with the described set of hyper-parameters for reproducability.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We discuss hyper-parameters in the paper along with the codebase for earier reproducability.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report all our analysis across 5 random runs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Detailed in appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: -

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This paper proposes a Probabilistic Slot Attention algorithm, whose goal is to achieve identifiable object-centric representations. The work primarily makes theoretical advancements in the field of object-centric learning, and as such it has little immediate societal or ethical consequences. Our method might be a step towards interpretable and aligned models which are desired properties of trustworthy AI.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: -

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: -

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: -

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: -

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: -

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.