
Variational Inference for Scalable 3D Object-centric Learning

Tianyu Wang

School of Computing

The Australia National University

tianyu.wang2@anu.edu.au

Kee Siong Ng

School of Computing

The Australia National University

keesiong.ng@anu.edu.au

Miaomiao Liu

School of Computing

The Australian National University

miaomiao.liu@anu.edu.au

Abstract

We tackle the task of scalable unsupervised object-centric representation learning on 3D scenes. Existing approaches to object-centric representation learning show limitations in generalizing to larger scenes as their learning processes rely on a fixed global coordinate system. In contrast, we propose to learn view-invariant 3D object representations in localized *object coordinate systems*. To this end, we estimate the object pose and appearance representation separately and explicitly map object representations across views while maintaining object identities. We adopt an amortized variational inference pipeline that can process sequential input and scalably update object latent distributions online. To handle large-scale scenes with a varying number of objects, we further introduce a *Cognitive Map* that allows the registration and query of objects on a per-scene global map to achieve scalable representation learning. We explore the object-centric neural radiance field (NeRF) as our 3D scene representation, which is jointly modelled within our unsupervised object-centric learning framework. Experimental results on synthetic and real datasets show that our proposed method can infer and maintain object-centric representations of 3D scenes and outperforms previous models.

1 Introduction

In recent years, 2D and 3D unsupervised object-centric learning has attracted increasing attention. While 2D object-centric learning methods [Eslami et al., 2016, Lin et al., 2020, Burgess et al., 2019, Crawford and Pineau, 2019, Locatello et al., 2020] aim to identify and segment objects within images, 3D methods aim to reconstruct complete 3D scene structures in an object-centric manner using RGB or RGBD observations [Li et al., 2020, Stelzner et al., 2021, Chen et al., 2021, Henderson and Lampert, 2020]. The ability to understand 3D surroundings in an object-centric way is crucial for high-level tasks such as relational reasoning and object manipulation. The majority of existing 3D methods assume that target scene scales are small enough to fit into the field of view (FOV) of a single camera and are centered at the origin of pre-defined global coordinate systems [Li et al., 2020, Stelzner et al., 2021, Chen et al., 2021]. As a result, these models fail to generalize to scenes beyond training set scale.

In this work, we aim to remove the small-scene assumption and to handle scene of potentially unbounded scales. As each camera view can only capture a limited local region of a scene, obtaining a comprehensive scene representation requires aggregating information from a potentially unknown

number of diverse views. To this end, we propose as solution a 3D object-centric learning pipeline termed *Scalable Online Object Centric network in 3D* (*SOOC3D*). Specifically, SOOC3D formulates object-centric learning as an online latent variable inference problem, explicitly models object poses and infers view-invariant object representations in localized *object coordinate systems*. To maintain object identities during the inference process, we introduce a scalable memory mechanism named *Cognitive Map*,¹ which can be used to register and query detected objects. Our proposed model is an unsupervised learning framework that is trained to reconstruct RGBD observations using object-compositional neural radiance field (NeRF). Previous works show that object-centric NeRF models commonly exhibit lower reconstruction quality compared to per-scene NeRFs optimised directly with SGD [Mildenhall et al., 2021, Zhang et al., 2022, Tancik et al., 2022]. This is due to the network bottlenecks filtering out high-frequency information [Engelcke et al., 2020]. We show that our framework supports a scalable per-object NeRF finetuning process which improves the reconstruction quality with preserved object identities.

Our contributions are summarised as follows. i) We propose, to the best of our knowledge, the first unbounded scalable generative-model-based unsupervised 3D object-centric learning framework. ii) We learn the explicit object locations and view-invariant object representations separately via the amortized variational inference framework to achieve scalable online updating. iii) To store a potentially unbounded number of detected objects for scalable inference, we introduce *Cognitive Map* separating object representations management from the inference process. iv) We demonstrate that the reconstruction quality can be further improved via our per-object NeRF finetuning process.

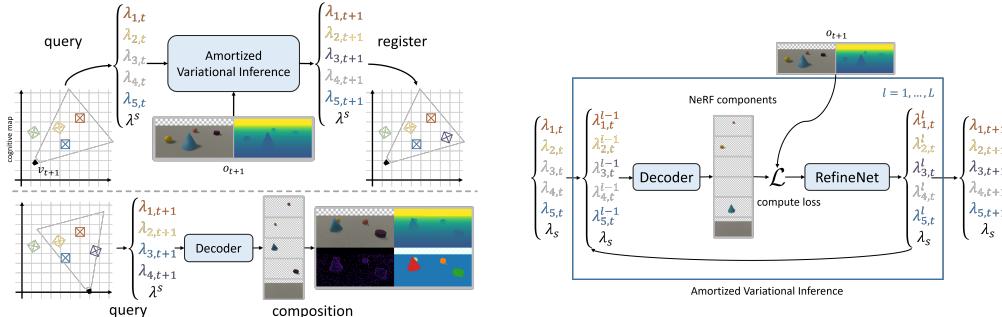


Figure 1: **Left:** The online inference process for scene updating (top) and novel view synthesis (bottom). Previously detected objects are registered in the cognitive map. Given a new observation, the representations of all objects in FOV are retrieved. Distributions over object latent variables and scene layout latent variables are parameterized by $\{\lambda_{i,t}\}$ and λ^s . If the number of objects existing in the current view is less than a pre-defined value K , we pad with priors (greyed $\lambda_{i,t}$). Amortized variational inference process updates $\{\lambda_{i,t}\}$ to integrate new information. Finally, we register updated $\{\lambda_{i,t+1}\}$ back into the cognitive map. For novel view synthesis, we sample latent variables and decode them into NeRF components. By composition, we obtain RGB, depth, segmentation and uncertainty map. **Right:** An L -iteration amortized variational inference process. In each iteration, the set of input representations is decoded into NeRF. The refinement network takes raw observation, reconstruction and other auxiliary variables to update latent variables.

2 Related Work

Unsupervised 2D Object-centric Learning. 2D object-centric learning aims to group pixels covering the same object under the same label and at the same time produce a neural representation of each discovered object. At the core of those methods is the spatial mixture model formulation that frames object-centric learning as a latent variable inference problem [Eslami et al., 2016, Greff et al., 2019, 2017]. To handle observations of high object density, a branch of works [Eslami et al., 2016, Crawford and Pineau, 2019, Lin et al., 2020] infers latent variables for local regions of each 2D observation. Pipelines equipped with iterative refinement modules [Greff et al., 2017, Locatello et al., 2020] refine the latent variable iteratively conditioned on an input view. Particularly, IODINE [Greff et al., 2019]

¹The term cognitive map is borrowed from cognitive psychology studies on mental representations of the spatial surroundings in animal, and human brain [Kitchin, 1994].

employs amortized variational inference [Marino et al., 2018] that can process sequential data. In dynamic scenes, motion cues can be exploited to improve the segmentation results [Singh et al., 2022, Karazija et al., 2022]. However, the aforementioned methods do not infer 3D structures. Object latents are discarded once out of view.

Unsupervised 3D Object-centric Learning. 3D-aware methods not only try to factorize observations in an object-centric manner but also infer the spatial structure of scenes, which can be examined by the means of novel views synthesis. Similar to its 2D counterpart, 3D object-centric representation learning approaches also adopt the spatial mixture model formulation. ObSuRF [Stelzner et al., 2021] and uORF [Yu et al., 2022] introduce neural radiance fields (NeRF) into the object-centric learning setting. Smith et al. [2022] propose to use object light field to avoid dense sampling along rays during rendering. As an attempt to model scenes on a larger scale, O3V [Henderson and Lampert, 2020] and SIMONe [Kabra et al., 2021] infer object-centric representation from a video sequence. However, both O3V and SIMONe adopt a non-incremental method and process entire video sequences before generating scene representations. MulMON [Li et al., 2020] adopts amortized variational inference to allow object latents to be updated by new views online. The methods mentioned above work well on scenes with sizes that can fit into the camera FOV but fail to scale up to larger scenes.

Object-Compositional NeRF and Scalable NeRF. Object-compositional NeRF has been studied recently to learn the 3D representation of each object and the scene for image synthesis. In particular, Yang et al. [2021] introduced a two-pathway framework to model the foreground objects and the scene branch, with known coarse object instance masks. Such a method cannot be directly applied to the unsupervised scenario. To handle large scenes, block-wise NeRF has been proposed recently[Zhang et al., 2022, Tancik et al., 2022]. They can either generalize to large scene [Zhang et al., 2022] by taking multiple images as input or train scene-specific block-wise NeRFs for large scene fast rendering [Zhang et al., 2022]. However, both Zhang et al. [2022] and Tancik et al. [2022] are not object aware. In this paper, we aim to combine the merits from all the approaches and learn scalable object-centric NeRF scene representations.

3 Method

We formulate object-centric representation learning as a latent variable inference problem with a dynamic latent variable set (Sec. 3.1 (a)). We then present a factorized variational proposal distribution tailored for scalable online inference (Sec. 3.1 (b)). Finally, we present implementation details of the proposed inference pipeline (Sec. 3.2).

3.1 Formulation and the Optimization Objective

(a) Generative Model. At each time t a camera of known pose v_t captures an RGB-D frame o_{v_t} of the scene. v_t is the extrinsic parameters of a camera in an arbitrary global coordinate system. Under the static scene assumption, each camera induces a fixed frame. Note that we do not pre-set an end time for data receiving, thus, t can be unboundedly large. To describe the observation received up to time t , we denote $\mathcal{O}_t = \{o_{v_1}, \dots, o_{v_t}\}$ and $\mathcal{V}_t = \{v_1, \dots, v_t\}$. We assume the entire scene is described by a set of latent variables $\mathcal{Z} = \{z_1, z_2, \dots\}$ of potentially unbounded size.

The exact latent variable posterior $p(\mathcal{Z}|\mathcal{O}_t, \mathcal{V}_t)$ is intractable. Considering that t is not assumed to be bounded, it is infeasible to process all views at once and an online approximate inference method is needed. Thus, we resort to the amortized variational inference framework [Greff et al., 2019, Li et al., 2020, Emami et al., 2021].

(b) Variational Inference.

We denote the latent variable set after observing \mathcal{O}_t and \mathcal{V}_t as $\mathcal{Z}_t = \{z_{1,t}, \dots, z_{m_t,t}\}$ and decompose latent variable $z_{i,t}$ as $z_{i,t} = (z_{i,t}^{where}, z_{i,t}^{what})$. $z_{i,t}^{what} \in \mathbb{R}^d$ is the object appearance embedding. While object poses in general have 6 degrees of freedom (DoF), under static scene assumption, 3 DoF is sufficient to achieve scalable inference. Thus, we only model its location on xz -plane and rotation about y -axis and set $z_{i,t}^{where} \in \mathbb{R}^3$. The other 3 DoF are modelled within object appearance embeddings implicitly.

Below, we show a way to approximate the exact posterior $p(\mathcal{Z}|\mathcal{O}_t, \mathcal{V}_t)$ with a parameterized proposal distribution $q(\mathcal{Z}_t|\mathcal{O}_t, \mathcal{V}_t)$. By exploiting the temporal and spatial structure of the problem, we simplify $q(\mathcal{Z}_t|\mathcal{O}_t, \mathcal{V}_t)$ to allow online and scalable inference.

Temporal factorization: First, we exploit the temporal structure of the problem and recursively factorize the proposal distribution as

$$q(\mathcal{Z}_t|\mathcal{O}_t, \mathcal{V}_t) = \int q(\mathcal{Z}_t|o_{v_t}, v_t, \mathcal{Z}_{t-1})q(\mathcal{Z}_{t-1}|\mathcal{O}_{t-1}, \mathcal{V}_{t-1})d\mathcal{Z}_{t-1}, \quad (1)$$

where $q(\mathcal{Z}_{t-1}|\mathcal{O}_{t-1}, \mathcal{V}_{t-1})$ is the posterior from the previous step and $q(\mathcal{Z}_t|o_{v_t}, v_t, \mathcal{Z}_{t-1})$ is the update distribution. This recursive factorization greatly simplifies the computation and allows us to update the posterior at each step by computing $q(\mathcal{Z}_t|o_{v_t}, v_t, \mathcal{Z}_{t-1})$ for unbounded t .

Spatial factorization: With $z_{i,t}^{where}$ explicitly modeled, we can exclude from the update distribution the latent variables that are out of the FOV of camera v_t by assuming that such latent variables will not contribute to the observation generation. For a latent set $\mathcal{Z}_{t'}$, we denote the set of latents in FOV of view v_t as $\mathcal{Z}_{t'}^{v_t}$ and its complement as $\bar{\mathcal{Z}}_{t'}^{v_t} = \mathcal{Z}_{t'} \setminus \mathcal{Z}_{t'}^{v_t}$ (the set of out-of-view latents). This spatial structure allows us to factorize the update distribution as

$$q(\mathcal{Z}_t|o_{v_t}, v_t, \mathcal{Z}_{t-1}) = q(\mathcal{Z}_{t'}^{v_t}|o_{v_t}, v_t, \mathcal{Z}_{t-1}^{v_t})q(\bar{\mathcal{Z}}_t^{v_t}|o_{v_t}, v_t, \bar{\mathcal{Z}}_{t-1}^{v_t}), \quad (2)$$

where $q(\bar{\mathcal{Z}}_t^{v_t}|o_{v_t}, v_t, \bar{\mathcal{Z}}_{t-1}^{v_t}) = \delta_{\bar{\mathcal{Z}}_{t-1}^{v_t}}(\bar{\mathcal{Z}}_t^{v_t})$ (Dirac delta). That is, latent variables that are not related to current observations remain unchanged. Now we can focus on the updated distribution of in-view latent sets $q(\mathcal{Z}_t^{v_t}|o_{v_t}, v_t, \mathcal{Z}_{t-1}^{v_t})$.

Surrogate distribution: $q(\mathcal{Z}_t^{v_t}|o_{v_t}, v_t, \mathcal{Z}_{t-1}^{v_t})$ is a distribution over a set. Both the set size and elements are random variables. A general solution is to generate set elements in an autoregressive manner, which is strictly sequential. For efficiency reason, we resort to parallelizable methods and assume that there can be at most K objects in each view during training. This assumption allows us to adopt a surrogate update distribution and parallelize the inference.

We define the augmentation of a latent variable $z_{i,t}$ as $\hat{z}_{i,t} = (z_{i,t}, z_{i,t}^{pres})$ with $z_{i,t}^{pres} \in \{0, 1\}$. $z_{i,t}^{pres}$ represents the existence of each object with $z_{i,t}^{pres} = 0$ indicating that the object does not exist. The augmentation of $\mathcal{Z}_{t-1}^{v_t}$ is defined as

$$\tilde{\mathcal{Z}}_{t-1}^{v_t} = \{\hat{z}_{i,t} | z_{i,t} \in \mathcal{Z}_{t-1}^{v_t}\} \bigcup \underbrace{\{\hat{z}_{0,0}, \dots, \hat{z}_{0,0}\}}_{K - |\mathcal{Z}_{t-1}^{v_t}|}, \quad (3)$$

where $\hat{z}_{0,0} := (z_{0,0}^{what}, z_{0,0}^{where}, z_{0,0}^{pres})$ is learnable global variational prior. Intuitively, if there are less than K elements in $\mathcal{Z}_{t-1}^{v_t}$ we pad with the global priors to K elements.

The surrogate update distribution is denoted as $q(\hat{\mathcal{Z}}_t^{v_t}|o_{v_t}, v_t, \tilde{\mathcal{Z}}_{t-1}^{v_t})$. $\hat{\mathcal{Z}}_t^{v_t}$ is a set of K augmented latent variables holding the updated latent variables. We factorize the surrogate distribution as

$$q(\hat{\mathcal{Z}}_t^{v_t}|o_{v_t}, v_t, \tilde{\mathcal{Z}}_{t-1}^{v_t}) = \prod_{\hat{z} \in \hat{\mathcal{Z}}_t^{v_t}} q(\hat{z}|o_{v_t}, v_t, \tilde{\mathcal{Z}}_{t-1}^{v_t}). \quad (4)$$

and implement it as a neural network. Finally, we recover the set of latent variables via

$$\mathcal{Z}_t^{v_t} = \{z_{i,t} | \hat{z}_{i,t} \in \hat{\mathcal{Z}}_t^{v_t}, z_{i,t}^{pres} = 1\}. \quad (5)$$

We parameterize $q(z_{i,t}^{where})$ and $q(z_{i,t}^{what})$ as isotropic Gaussian with parameter $\lambda_{i,t}^{where} = \{\mu_{i,t}^{where}, \sigma_{i,t}^{where}\}$ and $\lambda_{i,t}^{what} = \{\mu_{i,t}^{what}, \sigma_{i,t}^{what}\}$. $q(z_i^{pres})$ takes the form of Bernoulli distribution with $\lambda_{i,t}^{pres}$ being the logit. Continuous relaxation Maddison et al. [2017] is used for differentiable sampling. Scene layouts (floor) have modalities drastically different from objects. We define scene layout latent variables following the same definition but with fixed $z^{pres} = 1$ and pre-defined z^{where} Tancik et al. [2022]. For each view, one and only one scene layout variable is active.

Given the defined variational posterior above, we identify the core of the posterior inference to be $q(\hat{z}|o_{v_t}, v_t, \tilde{\mathcal{Z}}_{t-1}^{v_t})$ and we implement it via a refinement network f_ϑ . For each input view v_t , we

update our parametrized latent for L iterations. At iteration $l \in \{1, \dots, L\}$, the latent is updated as

$$\hat{z}_{i,t}^l \sim q_{\lambda_{i,t}^l}(\hat{z}_{i,t}^l) \quad (6)$$

$$\lambda_{i,t}^l = \lambda_{i,t}^{l-1} + f_\vartheta(\hat{z}_{i,t}^{l-1}, o_{v_t}, v_t, \mathbf{a}) \quad (7)$$

with $q_{\lambda_{i,t}^0} = q_{\lambda_{i,t-1}^L}$. That is, the refinement results of previous steps serve as the prior of the next step. The desired posterior $q(\hat{z}_{i,t} | \mathcal{O}_t, \mathcal{V}_t)$ is parameterized by $\lambda_{i,t}^L$. \mathbf{a} is a collection of auxiliary input. Latent variables are updated in parallel in each iteration. Specification of the auxiliary input and an algorithmic summary of the algorithm are provided in the supplementary.

(c) Training Objective. Below we use $\lambda_{:,t}^l$ to denote the parameters of all latent variables at time t after iteration l . Our training objective contains 3 terms. The input view observation log-likelihood is $\mathcal{L}^{input} = \sum_{t=1}^T \mathbb{E}_{q_{\lambda_{:,t}^L}} [\log p(o_{v_t} | v_t, \hat{\mathcal{Z}}_t^{v_t})]$, which is the sum of the log-likelihood of individual step. The KL term is $\mathcal{L}^{kl} = \sum_{t=1}^T \mathcal{D}_{KL}[q_{\lambda_{:,t}^L} || q_{\lambda_{:,t-1}^L}]$. The first two terms form the evidence lower bound (ELBO) of the intractable likelihood. To encourage the learning of view-invariant representation, in addition to the T input views, we also sample a set of query views \mathcal{Q} and compute the likelihood $\mathcal{L}^{query} = \sum_{(v,o_v) \in \mathcal{Q}} \mathbb{E}_{q_{\lambda_{:,t}^L}} [\log p(o_v | v, \hat{\mathcal{Z}}_T^v)]$. The training objective is

$$\mathcal{L} = \mathcal{L}^{input} - \mathcal{L}^{kl} + \mathcal{L}^{query} \quad (8)$$

By adopting the depth-informed NeRF likelihood function Stelzner et al. [2021], we are able to estimate the likelihood with two samples per ray. Details on training objectives are provided in the supplementary.

3.2 Model Implementation

The learning of view-invariant object appearance representation heavily relies on the pose-induced local coordinate system (a), within which we decode NeRF to reconstruct observation (b). Then we detail our refinement network implementation (c). The latent variable in-view test is implemented within a *Cognitive Map* data structure which also keeps track of all latent variables through the inference process (d). Then, we describe our curriculum learning setup (e) and how our per-object finetuning pipeline can be applied on top of our inference results (f). Below we denote retrieved latents as $\hat{z}_{\phi_v(k)}$ where $\phi_v(k)$ is the global index of the k^{th} element of $\hat{\mathcal{Z}}^v$ and omit subscript if clear from the context. The model pipeline is shown in Fig. 1.

(a) Object Coordinate System. Each $z_{\phi_v(k)}^{where}$ corresponds to a pose of an object component in the current camera coordinate system. For each component, we build a matrix $\Pi(z_{\phi_v(k)}^{where}) \in SE(3)$ to map each point x in the camera coordinate system to the local coordinate system $x_k = \Pi(z_{\phi_v(k)}^{where}) \cdot x$. Before decoding, each point is mapped into all object local coordinate systems. Note that x_k is a differentiable function of $z_{\phi_v(k)}^{where}$. Thus, all gradients to x_k flow through $z_{\phi_v(k)}^{where}$.

(b) Object-aware NeRF Decoding. Conditioned on $z_{\phi_v(k)}^{what}$, a NeRF decoder assigns each point x_k a raw density $\tilde{\sigma}_k(x_k) \in [0, 1]$ and a RGB value $\tilde{c}_k(x_k)$. Crucial to position learning, we introduce an inductive bias in the form of Gaussian weighting. To be more precise, we compute the weighted density $\log \hat{\sigma}_k(x_k) = \log \tilde{\sigma}_k(x_k) + \log w_g(x_k) - \mathcal{SG}(\log w_g(x_k)) + \log z_{\phi_v(k)}^{pres}$ where $w_g(\cdot)$ is a zero mean gaussian function and \mathcal{SG} is the stop gradient operation. By adding $\log w_g(x_k) - \mathcal{SG}(\log w_g(x_k))$, we encourage the $z_{\phi_v(k)}^{where}$ to be close to object center but the value of the weighted density remain unchanged. Weighted by $z_{\phi_v(k)}^{pres}$, non-existent components are turned off.

We then compute the normalized density as $\bar{\sigma}_k(x_k) = \frac{\hat{\sigma}_k(x_k)^2}{\sum_{i=0}^K \hat{\sigma}_i(x_i)}$. Note that $\sum_k \bar{\sigma}_k(x_k) \in [0, 1]$ allowing us to represent concrete object or void space. The final NeRF density at point x is given by $\sigma(x) = \sigma_{max} \cdot \sum_{k=0}^K \bar{\sigma}_k(x_k)$ with σ_{max} being the maximum NeRF density of our choice. The color is given by $c(x) = \sum_{k=0}^K \frac{\hat{\sigma}_k(x_k)}{\sum_{i=0}^K \hat{\sigma}_i(x_i)} \tilde{c}_k(x_k)$.

During training, it is sufficient to evaluate one sample on the surface and one sample in the air for each ray (see the supplementary for details). For testing, we densely sample the camera frustum, evaluate samples and compose via rendering equations.

(c) Refinement Network. At each time step t , give observation o_{v_t}, v_t , the refinement network implements the update distribution defined in Eq. 4. Following the amortized variational inference literature [Greff et al., 2019, Li et al., 2020], the refinement network takes as input latent variables and a set of auxiliary data and outputs the updated distributions. The auxiliary data may include observation, reconstruction, likelihood and gradient to latent variables, etc. Crucial to scalability, the refinement network interprets z^{where} as object locations in the camera coordinate system of v_t . That is, the refinement network is global coordinates and scene scale agnostic.

For each input view, the refinement process is executed in parallel for all components for L times. While new objects may be detected and old objects are out-of-view when cameras are switched, we do not hard code any object-matching heuristics. Similar to NeRF decoder, all object latent variables share one refinement network while the scene layout component has its own refinement network. Network structure and auxiliary input specifications are presented in the supplementary.

(d) Cognitive Map. A cognitive map stores all latent variables, implements the in-view test and interacts with the inference process via the registration and the query process.

Registration: After each refinement step, we register the updated latent variables. Recall that refinement results in a set of augmented latent variables. We first discard all \hat{z} with $q(z^{pres} = 1) < 0.5$ since they are deemed non-existent. Given the camera pose, all z^{where} are transformed from the camera coordinate system (where refinement happens) into the global coordinate system. Then \hat{z} are stored in a list for future queries.

Query: Before each refinement step, we query all in-view latent variables. Given a camera pose, we go through all registered latent variables and perform in-view checks. If less than K objects are found, pad with priors. For the first view in a scene, only priors are returned. In practice, it is possible to find more than K objects in one view. In this case, we retrieve those with top- $K \lambda^{pres}$ value. Then all retrieved z^{where} are transformed from the global coordinate system into the camera coordinates system for refinement.

The cognitive map only exposes relevant latent variables to the inference process. Thus, the inference memory consumption is independent of the total number of latent variables. During training, gradients of latent variables can flow through the registration-query loop. During testing, cognitive maps can be deployed on any storage.

(f) Per-Object NeRF Finetuning. We implement the per-object NeRF finetuning using the expectation-maximization algorithm. The latent variables are now the object identities of all ray samples. As initialization, we duplicate the trained NeRF decoder for each latent in the cognitive map. During the finetuning process, z^{where} is fixed and z^{what} is treated as part of per-object NeRF parameters. The evidence lower bound is maximized via direct gradient descent. Detailed formulations are provided in the supplementary. The finetuning process does not diminish the scalability as the per-object NeRF can also be registered into the cognitive map for future queries or finetuning.

4 Experiments

Dataset. The datasets commonly used in object-centric learning literature are often limited in scene scale, making them unsuitable for scalability studies [Eslami et al., 2018, Johnson et al., 2017, Engelcke et al., 2021, Yu et al., 2022]. Thus, we constructed two datasets consisting of multi-view RGBD data. The Unity dataset is created using Unity3D Juliani et al. [2020] and consists of simple geometry mimicking the object room dataset Eslami et al. [2018]. While the Unity dataset is limited in visual complexity, it contains scenes of three different scales termed as *small* (s), *medium* (m) and *large* (l). The *small* scenes are similar to those in previous datasets in that they can roughly fit into the field of view of a single camera. The *medium* scenes in this dataset are twice the size of the *small* scenes, and the *large* scenes are six times the size of the *small* scenes, making them particularly challenging to handle. The Blender dataset is created using Blender and features varying ground textures, non-trivial geometries, and photorealistic rendering. The Blender dataset also contains the three scales. See the supplementary for details on data generation and view sampling.

Metric. We render the inferred object-compositional NeRF into 2D RGB, depth and instance masks from each view. We report per-pixel root-mean-square-error (RMSE) on both RGB and depth value and compute mean-intersection-over-union (mIoU) score against ground truth masks. Pixels with depth values larger than the camera clipping depth are masked out. We additionally report the L2

distance between inferred object location and the ground truth location. We report quantitative results for both input (I) and query (Q) views.

Baseline. We compare our method with MulMON [Li et al., 2020], the state-of-the-art multi-view 3D scene object-centric learning method with online inference ability. We adopt their official implementation and additionally add depth as both input and reconstruction targets.

4.1 Scalable Object Centric Learning

In practice, training scenes are typically of a limited scale, while the scenes to be deployed may have varying sizes. To exam the test time scalability, we train the baseline and our model on small (MulMON_small, Ours_small) and medium (MulMON_medium, Ours_medium) scenes only and test them on all three scene scales. For our model, we fix K to 7 for all scene scales. For the baseline, we set the number of mixing components to be strictly larger than the total number of objects in the scenes during both training and testing. The quantitative results are reported in Table 1. The results of our approach are obtained as the average of 5 runs.

MulMON achieves 0.612 mIoU when trained and tested on small scenes. This setup aligns with their assumption that all objects appear in all views. However, when evaluated on medium and large-scale scenes, its performance drops to below 0.2 mIoU. Training on medium scenes does not improve performance.

By contrast, our model, being scene scale agnostic, can generalize to large scenes with a 0.07 mIoU performance drop when trained on small scenes. After training on medium scenes, our model generalizes to large scenes with no performance drop (see Fig. 2 for qualitative comparison). The comparable performances on the input and query views demonstrate that our model infers view-invariant features resulting in robust rendering from any view.

Table 1: Quantitative results on the Unity dataset.

	mIoU \uparrow			RGB RMSE \downarrow			depth RMSE \downarrow			L2 coord. error \downarrow		
	s	m	l	s	m	l	s	m	l	s	m	l
MulMON_small [Li et al., 2020]	I	0.612	0.198	0.158	0.055	0.167	0.178	0.430	0.972	1.183	N/A	
	Q	0.599	0.192	0.152	0.056	0.171	0.186	0.455	1.001	1.140	N/A	
MulMON_medium [Li et al., 2020]	I	0.371	0.225	0.141	0.102	0.141	0.144	0.891	0.991	1.223	N/A	
	Q	0.365	0.221	0.136	0.103	0.149	0.159	0.901	1.072	1.216	N/A	
Ours_small	I	0.763	0.721	0.694	0.074	0.107	0.141	0.516	0.680	0.741	N/A	
	Q	0.761	0.713	0.690	0.073	0.109	0.149	0.517	0.691	0.723	0.068	0.170 0.192
Ours_medium	I	0.710	0.756	0.761	0.075	0.098	0.091	0.617	0.650	0.634	N/A	
	Q	0.703	0.751	0.757	0.073	0.099	0.092	0.619	0.652	0.640	0.099	0.117 0.111

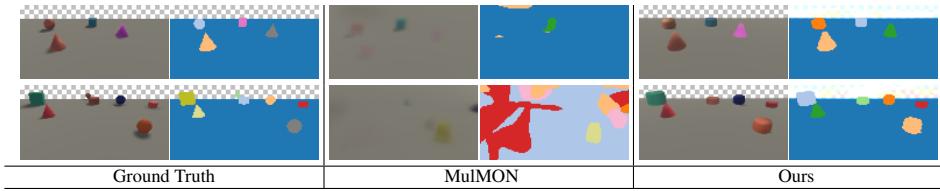


Figure 2: Query view synthesis in large scenes with models trained on medium scenes.

In Fig. 3 we visualize a 4-step online inference process. For each step, our model discovers new objects, updates the corresponding latent variables and registers them to the cognitive map. The inference process can repeat indefinitely to cover arbitrarily large areas. In the supplementary, we present additional qualitative results and further experiments exploring various aspects of our model.

To examine the ability to handle non-trivial geometries and higher visual complexity, we conduct experiments on the Blender dataset. We additionally apply per-object finetuning on top of the inference results. Quantitative results are reported in Table 2.

When dealing with complex geometries, our proposed model infers accurate object structures and outperforms the baseline in the small scene setup, while the baseline model tends to predict bubble-like shapes (Fig. 4). Additionally, after training on medium-sized scenes, our model generalizes to large scenes without any performance degradation, being scene scale agnostic. In contrast, the performance of the baseline model significantly deteriorates in large scenes (Fig. 4).

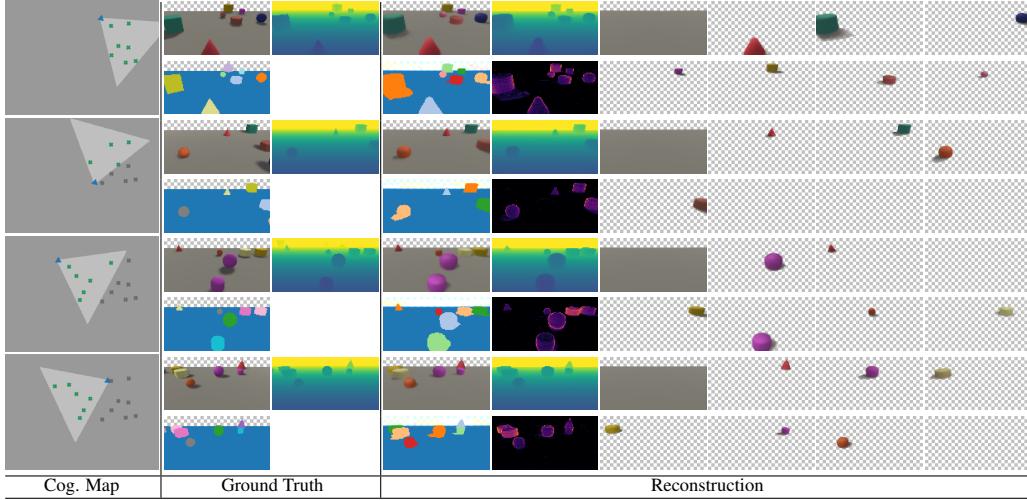


Figure 3: An online inference process from top to bottom. Each row corresponds to one update step. The left column shows the evolution of the cognitive map. The camera pose of each step is marked by a blue triangle and the camera cone (FOV) is highlighted. Each object latent registered in the cognitive map is marked with an x and is greyed out if outside of view.

We apply per-object finetuning to the Blender dataset inference results. As reported in Table 2 last row, the performance increases across all metrics. Object structure details are recovered with preserved object identities. Finetuned NeRFs can be registered into the cognitive for future queries.

Table 2: Quantitative results on the Blender dataset.

	mIoU \uparrow			RGB RMSE \downarrow			depth RMSE \downarrow			
	s	m	l	s	m	l	s	m	l	
MulMON_small [Li et al., 2020]	I	0.492	0.182	0.121	0.070	0.131	0.139	0.422	0.438	0.451
	Q	0.489	0.176	0.114	0.069	0.132	0.138	0.423	0.440	0.452
MulMON_medium [Li et al., 2020]	I	0.311	0.185	0.106	0.124	0.127	0.136	0.432	0.454	0.450
	Q	0.304	0.171	0.100	0.122	0.129	0.137	0.435	0.446	0.453
Ours_small	I	0.741	0.667	0.661	0.071	0.084	0.086	0.402	0.413	0.419
	Q	0.734	0.668	0.659	0.072	0.089	0.090	0.401	0.415	0.418
Ours_medium	I	0.699	0.684	0.682	0.088	0.086	0.082	0.409	0.411	0.414
	Q	0.698	0.673	0.678	0.085	0.083	0.084	0.407	0.412	0.416
Ours_finetune	I	0.861	0.848	0.830	0.026	0.025	0.023	0.248	0.251	0.259
	Q	0.858	0.839	0.826	0.031	0.029	0.028	0.250	0.254	0.264



Figure 5: Mesh reconstruction visualization. In each image from left to right is the SOOC3D inference result, SOOC3D+ (per-object finetuning) result, and ground truth mesh.

To evaluate the 3D object reconstruction quality, we extract meshes from NeRF components via marching cube and compute accuracy and completeness scores against ground truth meshes as defined in previous works Aanæs et al. [2016]. We report that without finetuning, our model achieves on average 7.8 cm accuracy and 5.4 cm completeness. After finetuning, we achieve on average 5.3 cm accuracy and 4.2 cm completeness. Note that this reconstruction result is obtained by only observing limited sparse views. As shown by the qualitative comparison in Fig. 5, the per-object finetuning process recovers fine details including extremely thin structures.

4.2 Real-world Dataset Results

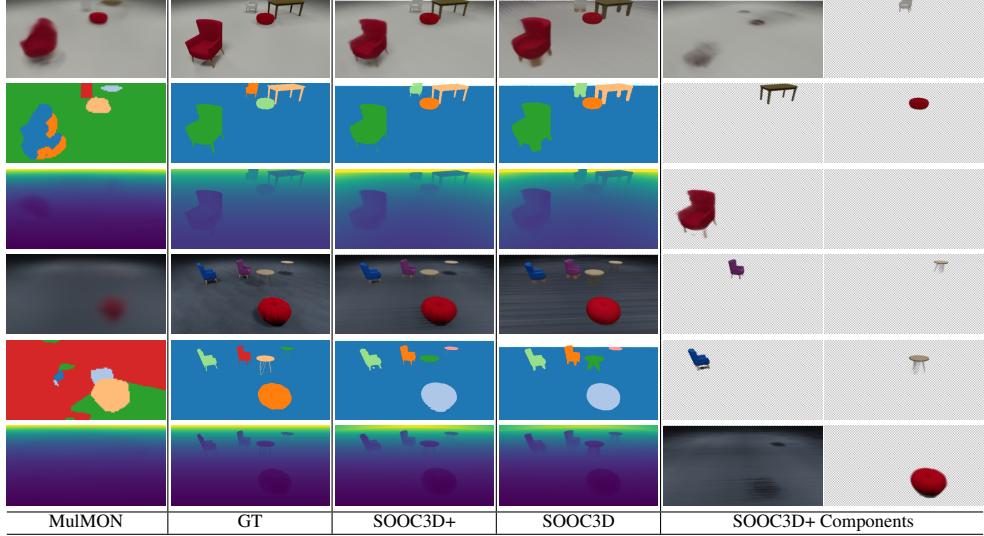


Figure 4: **Small scenes** (top three rows) and **large scenes** (bottom three rows) novel view synthesis produced by the baseline (MulMON), our model (SOOC3D), and per-object finetuning (SOOC3D+).

To demonstrate the potential of our method, we apply our model to Habitat-Matterport 3D (HM3D) Ramakrishnan et al. [2021], a real-world dataset comprised of textured meshes obtained from scanned indoor scenes. In our experiment, we specifically focus on segmenting salient furniture with regular structures, such as chairs, tables, and sofas. We thus render RGBD data from a set of 50 scenes within the dataset. The object segmentation performance is reported in Table 3. The baseline model [Li et al., 2020] struggled in inferring correct object identities due to the high visual complexity of scenes.

As shown in Fig. 6, our model learns to segment furniture and infer their basic structures. The per-object finetuning process significantly improves the segmentation quality with preserved object identities. We do observe that our model tends to overlook small-scale objects such as sofa cushions or floral displays on tables.

Table 3: HM3D segmentation results.

	mIoU
MulMON [Li et al., 2020]	0.134
SOOC3D	0.396
SOOC3D+	0.521

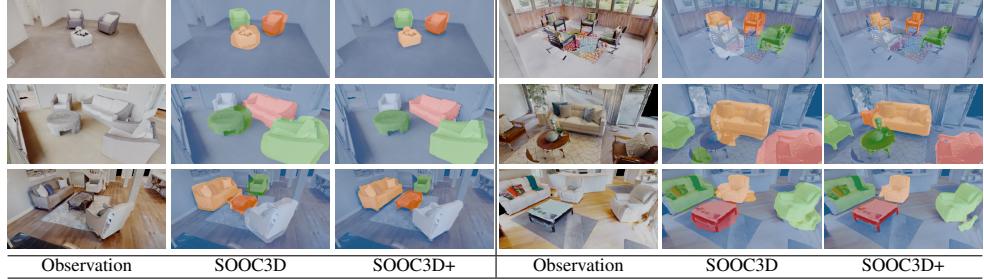


Figure 6: Predicted furniture instance masks on HM3D dataset.

5 Limitation

Our pipeline aims to recover 3D scene geometry and enable scalable inference. However, there are two limitations to consider. First, we only model 3 DoF for object poses instead of 6. As a result, the appearance embedding of an object in a standing position differs from that of the same object in a lying down position. Second, our model is designed for handling a static scene and is not suitable for handling moving objects. To overcome these limitations, one can use the same variational inference formulation and incorporate a transition model to predict 6 DoF object motions between time steps.

6 Conclusion and Future Work

We propose a framework for unsupervised 3D object-centric learning for handling scenes of large scale and a varying number of objects in the scene. We introduced factorized latent learning which separates the object pose and view-invariant appearance latent variables. Our object-compositional NeRF allows the learning of 3D representation in the object coordinate system. The cognitive map ensures object permanence and keeps track of all detected objects. The inference results on HM3D demonstrate the potential of 3D object-centric learning algorithms. Our learned view-invariant 3D object representation can potentially be applied in the SLAM system or relational reasoning tasks. In future work, we aim to achieve 6 DoF object pose estimation and dynamic scene modeling.

References

- S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS’16*, page 3233–3241, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.
- 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. In *ICLR*, 2020. URL <https://openreview.net/forum?id=rk103ySYDH>.
- Christopher Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *ArXiv*, abs/1901.11390, 01 2019. URL <https://arxiv.org/abs/1901.11390>.
- Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. *AAAI*, 33:3412–3420, 07 2019. doi: 10.1609/aaai.v33i01.33013412.
- Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020.
- Nanbo Li, Cian Eastwood, and Robert Fisher. Learning object-centric representations of multi-object scenes from multiple views. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5656–5666. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/file/3d9dabe52805a1ea21864b09f3397593-Paper.pdf>.
- Karl Stelzner, Kristian Kersting, and Adam R. Kosiorek. Decomposing 3d scenes into objects via unsupervised volume segmentation, 2021.
- Chang Chen, Fei Deng, and Sungjin Ahn. Roots: Object-centric representation and rendering of 3d scenes, 2021.
- Paul Henderson and Christoph H. Lampert. Unsupervised object-centric video generation and decomposition in 3D. In *NeurIPS*, 2020.
- Rob Kitchin. Cognitive maps: What are they and why study them? *Journal of Environmental Psychology*, 14:1–19, 03 1994. doi: 10.1016/S0272-4944(05)80194-X.
- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, dec 2021. ISSN 0001-0782. doi: 10.1145/3503250. URL <https://doi.org/10.1145/3503250>.
- Xiaoshuai Zhang, Sai Bi, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Nerfusion: Fusing radiance fields for large-scale scene reconstruction. *arXiv preprint arXiv:2203.11283*, 2022.
- Matthew Tancik, Vincent Casser, Xincheng Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. *arXiv preprint arXiv:2202.05263*, 2022.
- Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. Reconstruction Bottlenecks in Object-Centric Generative Models. *ICML Workshop on Object-Oriented Learning*, 2020.
- Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loïc Matthey, Matthew M Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *ICML*, 2019.
- Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *NeurIPS, NIPS’17*, page 6694–6704, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

- Joseph Marino, Yisong Yue, and Stephan Mandt. Iterative amortized inference. In *ICML*, 07 2018.
- Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL <https://openreview.net/forum?id=eYfIM8MTUE>.
- Laurynas Karazija, Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Unsupervised multi-object segmentation by predicting probable motion patterns. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_w2-1nXNjvv.
- Hong-Xing Yu, Leonidas Guibas, and Jiajun Wu. Unsupervised discovery of object radiance fields. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=rwE8SshAlxw>.
- Cameron Smith, Hong-Xing Yu, Sergey Zakharov, Fredo Durand, Joshua B. Tenenbaum, Jiajun Wu, and Vincent Sitzmann. Unsupervised discovery and composition of object light fields, 2022. URL <https://arxiv.org/abs/2205.03923>.
- Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matthew Botvinick, Alexander Lerchner, and Christopher P. Burgess. SIMONe: View-invariant, temporally-abstacted object representations via unsupervised video decomposition. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL <https://openreview.net/forum?id=YSzTMntO1KY>.
- Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13779–13788, 2021.
- Patrick Emami, Pan He, Sanjay Ranka, and Anand Rangarajan. Efficient iterative amortized inference for learning symmetric and disentangled multi-object representations. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2970–2981. PMLR, 18–24 Jul 2021. URL <http://proceedings.mlr.press/v139/emami21a.html>.
- Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=S1jE5L5g1>.
- S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. ISSN 0036-8075. doi: 10.1126/science.aar6170. URL <https://science.sciencemag.org/content/360/6394/1204>.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1988–1997, 07 2017. doi: 10.1109/CVPR.2017.215.
- Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. GENESIS-V2: Inferring Unordered Object Representations without Iterative Refinement. *arXiv preprint arXiv:2104.09958*, 2021.
- A. Juliani, V. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, and D. Lange. Unity: A general platform for intelligent agents. *ArXiv*, abs/1809.02627, 2020.
- Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjørholm Dahl. Large-scale data for multiple-view stereopsis. *Int. J. Comput. Vision*, 120(2):153–168, nov 2016. ISSN 0920-5691. doi: 10.1007/s11263-016-0902-9. URL <https://doi.org/10.1007/s11263-016-0902-9>.

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL <https://arxiv.org/abs/2109.08238>.