Meta-probabilistic Modeling

Kevin Zhang
Massachusetts Institute of Technology

Yixin Wang
University of Michigan

Abstract

Probabilistic graphical models (PGMs) are widely used to discover latent structure in data, but their success hinges on selecting an appropriate model design. In practice, model specification is difficult and often requires iterative trial-and-error. This challenge arises because classical PGMs typically operate on individual datasets. In this work, we consider settings involving collections of related datasets and propose *meta-probabilistic modeling (MPM)* to learn the generative model structure itself. MPM uses a hierarchical formulation in which global components encode shared patterns across datasets, while local parameters capture dataset-specific latent structure. For scalable learning and inference, we derive a tractable VAE-inspired surrogate objective together with a bi-level optimization algorithm. Our methodology supports a broad class of expressive probabilistic models and has connections to existing architectures, such as Slot Attention. Experiments on object-centric representation learning and sequential text modeling demonstrate that MPM effectively adapts generative models to data while recovering meaningful latent representations.

1 INTRODUCTION

Probabilistic graphical models (PGMs) provide a principled method for discovering and analyzing latent structure in data (Silva and Ghahramani, 2009; Li et al., 2013). PGMs represent the underlying generative process using graphical structures that encode dependencies among random variables, which can enable efficient learning and inference (Jordan et al., 1999; Wainwright and Jordan, 2008). This framework encompasses a broad family of models, ranging from latent variable models for hierarchical organization (Bishop and Tipping, 1998) to state-space models for capturing temporal dynamics (Rabiner and Juang, 1986; Doerr et al., 2018). Because of their expressiveness and interpretability, PGMs are widely applied across numerous domains, including semantic topic discovery in natural language processing (Blei et al., 2003; Blei and Lafferty, 2006; Blei, 2012) and molecular interaction modeling in computational biology (Airoldi, 2007).

However, the effectiveness of PGMs depends on selecting well-specified models that appropriately capture the underlying dependencies and structure of the data (Koller and Friedman, 2009). This typically requires the practitioner to iteratively refine both the

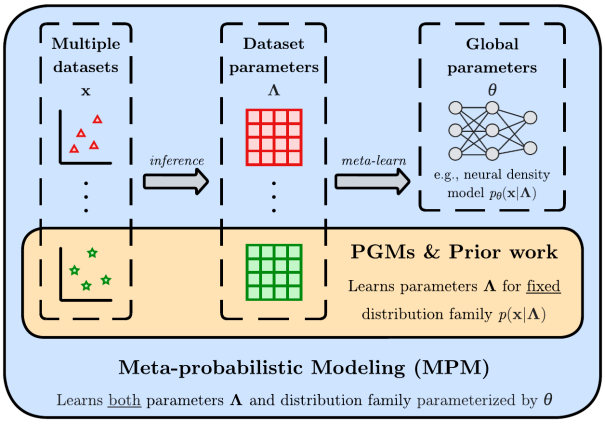


Figure 1: MPM infers dataset parameters Λ from multiple datasets $\mathbf{x}$, and meta-learns the conditional distribution $p_\theta(\mathbf{x} \mid \Lambda)$ using a (potentially neural) parameterization. Unlike classical PGMs and most prior work, which typically assume a fixed model family (e.g., Gaussian), MPM jointly learns both the distribution family and its parameters from data.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

graphical structure and the associated distributional families. Designing the model by hand rather than learning it from data can lead to misspecification, including imprecise modeling of observations and latent components, rigid graphical topologies that fail to adapt to heterogeneous data, and structural assumptions that may be overly restrictive (Juang and Rabiner, 1985; Kingma and Welling, 2013).

Meta-probabilistic modeling. To address these challenges, we introduce *meta-probabilistic modeling (MPM)*, an approach for learning the model directly from data. Because PGMs usually operate at the level of a single dataset, the underlying form is under-specified. Our approach instead leverages a collection of datasets sharing related latent structure to infer the model, as shown in Figure 1.

We posit a hierarchical architecture with *global parameters* shared across datasets and *dataset-specific parameters* that capture variation at the dataset and local level. The global components specify the form of the distributional families and are parameterized flexibly (e.g., using neural networks), while the local components model the underlying latent structure. This design combines the interpretability of probabilistic modeling with the expressive capacity of deep learning architectures.

As with most latent variable models, computing the posterior distribution over the model parameters in our method is intractable and must be approximated (Wainwright and Jordan, 2008; Koller and Friedman, 2009). We show that inference in meta-probabilistic modeling can remain tractable, even when parts of the generative process are parameterized by a complex family, such as neural networks. At a high level, we construct a surrogate potential inspired by recognition networks in variational autoencoders (VAEs), which enables analytic local coordinate ascent updates for dataset-specific parameters, while learning the global generative components via gradient-based optimization.

Contributions. Our main contributions are as follows: (1) we propose *meta-probabilistic modeling (MPM)*, a probabilistic framework that learns generative model structure across multiple related datasets by combining hierarchical PGMs with flexible global parameterizations; (2) we derive an efficient and scalable learning algorithm with principled connections to Slot Attention (Locatello et al., 2020); and (3) we validate MPM on object-centric representation learning and sequential text modeling, where it recovers meaningful latent structure while flexibly adapting to complex data.

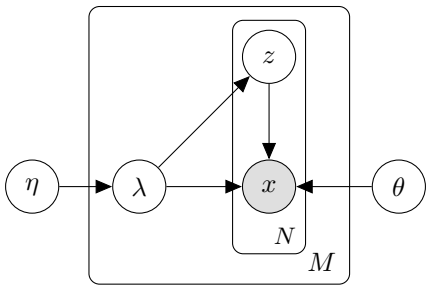


Figure 2: The latent variable meta-probabilistic model separates global and dataset parameters hierarchically: θ indexes the data generating distribution family $p_\theta(x \mid z, \lambda)$, and η defines a prior over λ.

Note that while we parameterize the shared components using neural networks, our method itself is agnostic to the specific architectures employed. Rather, our primary aim in this work is to articulate a broad class of rich, probabilistic models for multi-dataset settings and to derive a concrete learning and inference procedure.

2 META-PROBABILISTIC MODELING

In this section, we formalize MPM and our corresponding learning and inference algorithm.

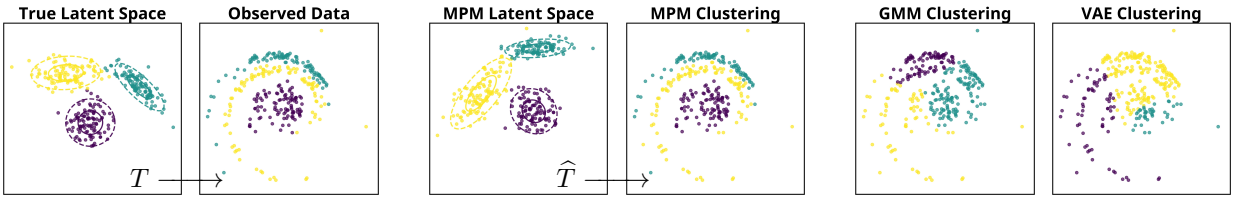
2.1 Problem formulation

We consider settings with a collection of M related datasets $\{\mathcal{D}_i\}_{i=1}^M$, where each dataset $\mathcal{D}_i = \{x_{ij}\}_{j=1}^{N_i}$ consists of (potentially high-dimensional) observations from an underlying latent variable model. To model this structure, we introduce latent *local parameters* z_{ij} for individual datapoints, *dataset parameters* $\Lambda = \{\lambda_i\}_{i=1}^M$ that encode variation across datasets, and *global parameters* η, θ that govern the generative process of the dataset parameters and observations, respectively. This hierarchical setup differs from most previous work, which do not specify separate levels for global and dataset parameters (Krishnan et al., 2017; Johnson et al., 2016). A graphical model representation of this framework is depicted in Figure 2.

The objective is to learn the global and dataset parameters that maximize the data likelihood:

$$\mathcal{L}(\Lambda, \theta, \eta) \coloneqq \sum_i \mathcal{L}_i(\Lambda, \theta, \eta),$$

$$\mathcal{L}_i(\Lambda, \theta, \eta) \coloneqq \log p_\eta(\lambda_i) + \sum_j \log p_\theta(x_{ij} \mid \lambda_i).$$



(a) (b) (c)

Figure 3: In our example, where the observed data is generated by applying a spiral-shaped deformation to the underlying latent representation (Fig. 3a, left), MPM successfully learns the non-linear transformation T using multiple datasets. The resulting cluster assignments (3b, right) more closely match the ground truth (3a, right) compared to those from a canonical GMM or VAE (3c).

Like most latent variable models, the true posterior over z_{ij} is generally intractable. One standard solution to this problem uses variational inference, which posits an approximate posterior q and maximizes the Evidence Lower Bound (ELBO) $\mathcal{L}^{\text{ELBO}} \leq \mathcal{L}$, where $\mathcal{L}^{\text{ELBO}} \coloneqq \sum_i \mathcal{L}_i^{\text{ELBO}}$, and

$$\mathcal{L}_i^{\text{ELBO}} \coloneqq \log p_\eta(\lambda_i) + \sum_j \mathbb{E}_q \left[\log \frac{p_\theta(x_{ij}, z_{ij} \mid \lambda_i)}{q(z_{ij})} \right]$$
$$= \log p_\eta(\lambda_i) + \sum_j \mathbb{E}_q \left[\log \frac{p_\theta(x_{ij} \mid z_{ij}, \lambda_i) p(z_{ij} \mid \lambda_i)}{q(z_{ij})} \right].$$

Here, $\mathcal{L}^{\text{ELBO}}$ and $\mathcal{L}_i^{\text{ELBO}}$ are implicitly understood to be functions of Λ, θ, η, and q.

2.2 Spiral GMM example

To illustrate the motivation behind MPM, consider a simple toy example involving Gaussian Mixture Models (GMMs). Suppose a practitioner wants to cluster data that originates from a mixture of Gaussians, but the observations have been transformed by an unknown mapping $x \mapsto Tx$ that potentially distorts the underlying cluster geometry.

In our example, we choose T to have the form $(r, \theta) \mapsto (r, \theta + \alpha r + \beta)$ in polar coordinates. This transformation produces the spiral-shaped dataset depicted in Figure 3a, where each cluster corresponds to an arm or the center of the spiral. The parameters α and β control the deformation strength and global angular offset, respectively.

Inferring the transformation T from a single dataset is under-specified. However, when multiple datasets are available and each transformed by the same mapping, MPM can exploit shared structure to learn a coherent latent representation by estimating the mapping $\widehat{T}$. As illustrated in Figure 3b, MPM recovers the latent space up to the global rotation β, yielding cluster assignments that match the ground truth. In contrast, a standard GMM in Figure 3c fails to recover the original cluster structure.

Note that learning the transformation T is equivalent to learning an induced distance function on the data. More generally, when a specific parameterization of T is unavailable, we can instead model it using an expressive function class, such as a neural network. In such cases, we demonstrate that the model remains tractable by optimizing an appropriate surrogate objective.

2.3 Fast and scalable inference using surrogate objectives

In classical PGMs, optimization of $\mathcal{L}_i^{\text{ELBO}}$ is typically carried out using coordinate-ascent procedures such as variational EM. However, this approach for MPM poses two computational challenges: (i) the number of parameters grows linearly with the number of datasets, since each dataset introduces λ_i and z_{ij}, and (ii) optimizing q becomes computationally expensive for complex models, because it requires repeated evaluation of $p_\theta(z_{ij} \mid x_{ij}, \lambda_i)$.

To address these issues, we define q and Λ implicitly as functions of the global parameters θ and η, by treating them as local partial optimizers of the ELBO. Solving this optimization is still intractable in general, so we instead introduce a tractable surrogate objective $\widehat{\mathcal{L}}_i^{\text{ELBO}}$ that admits efficient updates of q and Λ. The surrogate objective takes the form:

$$\widehat{\mathcal{L}}_i^{\text{ELBO}}(\Lambda, \phi, \eta, q) \coloneqq \log p(\lambda_i \mid \eta) + \sum_j \mathbb{E}_q \left[\log \frac{\exp\{\psi_\phi(z_{ij} \mid x_{ij}, \lambda_i)\} p(z_{ij} \mid \lambda_i)}{q(z_{ij})} \right],$$

Here, ψ_ϕ is a surrogate potential from a recognition network parameterized by ϕ, analogous to the inference network in variational autoencoders (VAEs).

Algorithm 1 Training and inference procedure for meta-probabilistic modeling

Require: Datasets $\{\mathcal{D}_i\}_{i=1}^M$, inner optimization steps T, minibatch size B, learning rate α
Output: Parameters $\lambda^0, \theta, \phi, \eta$

1: **Initialize** $\vartheta = \{\lambda^0, \theta, \phi, \eta\}$
2: **while** not converged **do**
3: Sample minibatch $\mathcal{B} \subseteq \{\mathcal{D}_i\}_{i=1}^M$ with $|\mathcal{B}| = B$
4: **Initialize** $\Lambda_{\phi,\eta}^{(0)} \leftarrow \{\lambda^0\}_{i=1}^M$
5: **for** $t = 1$ **to** T **do**
6: $q_{\phi,\eta}^{(t)} \leftarrow \arg\max_q \ \widehat{\mathcal{L}}_{\mathcal{B}}^{\text{ELBO}}\left(\Lambda_{\phi,\eta}^{(t-1)}, \phi, \eta, q\right)$
7: $\Lambda_{\phi,\eta}^{(t)} \leftarrow \arg\max_\Lambda \ \widehat{\mathcal{L}}_{\mathcal{B}}^{\text{ELBO}}\left(\Lambda, \phi, \eta, q_{\phi,\eta}^{(t)}\right)$
8: **end for**
9: $\mathcal{L}_{\mathcal{B}}^{\text{MPM}} \leftarrow \mathcal{L}_{\mathcal{B}}^{\text{ELBO}}\left(\Lambda_{\phi,\eta}^{(T)}, \theta, \eta, q_{\phi,\eta}^{(T)}\right)$
10: $\vartheta \leftarrow \texttt{SGD}(\vartheta, \nabla_\vartheta \mathcal{L}_{\mathcal{B}}^{\text{MPM}}, \alpha)$
11: **end while**
12: **return** $\lambda^0, \theta, \phi, \eta$

This construction is inspired by Johnson et al. (2016), which exploit conjugacy to obtain closed-form updates for q. For MPM, we instead choose ψ_ϕ such that q and Λ are jointly optimizable via a coordinate ascent procedure.

To make the dependence on the global parameters explicit, we define the following iterative updates:

$$\Lambda_{\phi,\eta}^{(t+1)} = \arg\max_{\Lambda} \widehat{\mathcal{L}}_i^{\text{ELBO}}(\Lambda, \phi, \eta, q_{\phi,\eta}^{(t)}),$$
$$q_{\phi,\eta}^{(t+1)} = \arg\max_{q} \widehat{\mathcal{L}}_i^{\text{ELBO}}(\Lambda_{\phi,\eta}^{(t+1)}, \phi, \eta, q),$$

with a shared initialization $\Lambda_{\phi,\eta}^{(0)} = \{\lambda^0\}_{i=1}^M$. The meta-probabilistic modeling objective is defined as

$$\mathcal{L}^{\text{MPM}}(\lambda^0, \theta, \phi, \eta) \coloneqq \sum_i \mathcal{L}_i^{\text{ELBO}}(\Lambda_{\phi,\eta}^T, \theta, \eta, q_{\phi,\eta}^T).$$

The MPM objective provides a lower bound on the data likelihood, in the sense that

$$\begin{aligned}\mathcal{L}(\Lambda, \theta, \eta) &\geq \max_{\Lambda, q} \sum_i \mathcal{L}_i^{\text{ELBO}}(\Lambda, \theta, \eta, q) \\ &\geq \sum_i \mathcal{L}_i^{\text{ELBO}}(\Lambda_{\phi,\eta}^T, \theta, \eta, q_{\phi,\eta}^T) = \mathcal{L}^{\text{MPM}}.\end{aligned}$$

The model is trained using a bi-level optimization procedure outlined in Algorithm 1. In the inner loop (lines 5-8), we optimize over q and Λ, while holding the generative model and recognition network fixed. The outer meta-learning step (lines 9-10) then updates the global parameters θ, η, recognition network parameters ϕ, and learnable initialization λ^0.

To ensure tractable inference, the surrogate potential ψ_ϕ must be chosen so that the inner optimization can be done efficiently. In such cases, optimizing over ϕ is effectively learning a recognition network that best approximates the posterior under the chosen ψ_ϕ. This approximation serves to make learning dataset variables computationally feasible in terms of time and parameter count.

Moreover, we emphasize that our proposed learning algorithm is not intrinsic to the MPM methodology itself. While we present a simple and scalable optimization procedure, alternative inference approaches, such as MCMC, could also be used.

3 TWO MPM CASE STUDIES

Our method requires access to multiple related datasets in order to learn both global and dataset-specific parameters. In this section, we consider two concrete and practical settings in which this assumption naturally arises, focusing on object-centric learning and sequential text modeling.

3.1 MPM for object-centric learning

We first apply MPM to object-centric learning, which aims to identify coherent objects within an image. We cast this task as a clustering problem in which pixels are grouped according to their semantic roles. This setting is well suited for MPM because global parameters model visual generative components shared across images, while dataset and local parameters capture the composition and arrangement of objects within each individual image.

Formally, we treat each image as a dataset $\mathcal{D}_i$, where $\{x_{ij}\}_{j=1}^{N_i}$ represents its pixels. Because the goal of object-centric learning is to cluster pixels within each image into semantically coherent objects, we accordingly model the dataset parameters $\lambda_i = \{\mu_{ik}\}_{k=1}^K$ as the K cluster centers of an image-specific GMM and z_{ij} as the cluster assignments. The centers are themselves generated from a global GMM with centers $\eta = \{\nu_\ell\}_{\ell=1}^L$. For simplicity, we assume isotropic Gaussian components with identity covariance and uniform mixing weights.

Under this formulation, our method uses a *mixture generative model* in which the local cluster centers are mapped through a learned transformation f_θ specified by global parameters.

$$p_\theta(x_{ij} \mid z_{ij} = k, \lambda_i) \propto \exp\left(-\frac{1}{2}\|x_{ij} - f_\theta(\mu_{ik})_j\|^2\right).$$

Here, f_θ can be viewed as a neural network that induces a learned distance function between the pixels x_{ij} and cluster centers μ_{ik}.

We define the surrogate potential to be,

$$\psi_\phi(z_{ij} = k \mid x_{ij}, \lambda_i) = -\frac{1}{2}\|\mu_{ik} - g_\phi(x_{ij})\|^2.$$

Returning to the VAE analogy, g_ϕ is a recognition network that maps each pixel into a shared latent space, while f_θ acts as a deterministic image decoder.

Additionally, we also consider an *additive generative model* with the form

$$p_\theta(x_{ij} \mid \lambda_i) \propto \exp\left(-\frac{1}{2}\left\|x_{ij} - \sum_k w(\lambda_i)_{jk} \cdot f_\theta(\mu_{ik})_j\right\|^2\right),$$

where w are alpha masks, normalized across clusters for each pixel. This additive decoder is commonly used in object-centric learning, in part due to its favorable theoretical identifiability properties (Greff et al., 2019; Lachapelle et al., 2023). We also consider this model in order to enable direct comparison with existing object-centric learning methods (Locatello et al., 2020; Wang et al., 2023).

Intuitively, both models fit an image-specific GMM in a shared latent space, with a prior over the cluster centers. This design admits closed-form update steps for q and Λ using coordinate ascent.

Proposition 1. *For fixed ϕ and η, the following updates for q and Λ (Algorithm 1, lines 6-7) do not decrease the surrogate ELBO.*

$$q(z_{ij} = k) \propto \exp\left(-\frac{1}{2}\|\mu_{ik} - g_\phi(x_{ij})\|^2\right),$$

$$\mu_{ik} = \frac{\sum_\ell r_{ik\ell}\nu_\ell + \sum_j s_{ijk} g_\phi(x_{ij})}{\sum_\ell r_{ik\ell} + \sum_j s_{ijk}},$$

where,

$$r_{ik\ell} = \frac{\exp\left(-\frac{1}{2}\|\mu_{ik} - \nu_\ell\|^2\right)}{\sum_{\tilde{\ell}} \exp\left(-\frac{1}{2}\|\mu_{ik} - \nu_{\tilde{\ell}}\|^2\right)},$$

$$s_{ijk} = \frac{\exp\left(-\frac{1}{2}\|\mu_{ik} - g_\phi(x_{ij})\|^2\right)}{\sum_{\tilde{k}} \exp\left(-\frac{1}{2}\|\mu_{i\tilde{k}} - g_\phi(x_{ij})\|^2\right)}.$$

The proof is provided in Appendix A. Since these updates can be computed efficiently, the optimization is tractable and can be carried out using Algorithm 1.

Connection with Slot Attention. Many modern architectures for object-centric learning build on the Slot Attention module introduced by Locatello et al. (2020). We show that our proposed approach for this task is closely related to Slot Attention and can be viewed through a similar modeling lens.

At a high level, the Slot Attention model encodes each image to a latent representation z, and a set of K slots s is iteratively refined using an attention mechanism. Each slot is intended to represent a distinct object in the scene. A decoder then maps each slot to an object-specific representation, which are then combined additively.

The primary contribution of Slot Attention is the slot refinement algorithm, which iteratively updates the set of slots by computing scaled dot-product attention between the latent representation and the slots. Let W_q, W_k, and W_v be the query, key, and value projection matrices, respectively, and let $s^{(t)}$ represent the slot embeddings at iteration t. The update at each step is given by:

$$A^{(t)} = \mathsf{Softmax}\left(\frac{(W_k z)(W_q s^{(t-1)})^T}{\sqrt{D}}, \mathsf{axis} = \mathsf{slots}\right),$$

$$u^{(t)} = \mathsf{WeightedMean}(\mathsf{weights} = A^{(t)}, \mathsf{values} = W_v z),$$

$$s^{(t)} = \mathsf{SlotUpdate}(u^{(t)}, s^{(t-1)}).$$

The initial slots $s^{(0)}$ are sampled from a learned Gaussian distribution and iteratively refined over T rounds. Additional details of the Slot Attention algorithm are provided in Appendix B.

Slot Attention shares several structural similarities with our meta-probabilistic model for object-centric learning. In particular, the dataset-level parameters Λ correspond to the slots, while the global parameters θ define the decoder. When viewed through this lens, the slot refinement procedure closely resembles the inner optimization steps of Algorithm 1 (lines 6–7). The attention weights $W^{(t)}$ correspond to the approximate posterior q, with similarity measured using a scaled dot-product attention instead of Euclidean distance, while the weighted mean update $u^{(t)}$ is a special case when $r_{ik\ell} = 0$.

Hence, Slot Attention admits a precise probabilistic interpretation by treating the iterative updates as approximate likelihood maximization in a latent clustering model. From this perspective, the effectiveness of Slot Attention primarily arises from its implicit role as a meta-probabilistic model, rather than from the attention mechanism itself.

The connection to MPM also provides a principled foundation for extending Slot Attention. In particular, we have considered a setting in which the slots themselves are generated by a global GMM parameterized by η. This extension enables the model to learn object-centric representations while simultaneously clustering the objects themselves to discover shared features across datasets.

3.2 MPM for sequential text modeling

We extend the idea of clustering pixels in images to text by grouping words within a document to uncover underlying semantic and syntactic themes. In this setting, we consider a corpus of documents, where each document corresponds to a dataset $\mathcal{D}_i$. The observations $x_{ij} \in \{1, \dots, V\}$ are the words in document i, where V denotes vocabulary size. The dataset parameters $\lambda_i = \{\mu_{ik}\}_{k=1}^K$ represent K latent topic embeddings.

We posit the following generative model:

$$p(z_{ij} = k \mid \lambda_i) = 1/K,$$
$$p(x_{ij} = v \mid z_{ij} = k, \lambda_i) = \mathsf{Softmax}(f_\theta(\mu_{ik}, s_j))_v$$

where f_θ maps a topic embedding μ_{ik} together with a positional encoding s_j to a distribution over words. We define the surrogate potential

$$\psi_\phi(z_{ij} = k \mid x_{ij}, \lambda_i) = -\frac{1}{2}\|\mu_{ik} - g_\phi(x_{ij})\|^2,$$

where g_ϕ is a recognition network that produces contextual embeddings for each word in document i. Because we reuse the surrogate potential from the object-centric learning setting, the resulting updates for q and Λ are identical. Our formulation corresponds to fitting a document-specific GMM in a shared latent space, with each mixture component representing a latent topic or theme.

Remark. So far, we have presented applications of MPM while keeping the models f_θ and g_ϕ abstract, imposing no restrictions on their specific forms. Our methodology is agnostic to the choice of these components and allows for any suitable high-capacity parameterization appropriate to the domain. For example, in object-centric learning we use standard CNN architectures, whereas in the text modeling setting, we use a pre-trained language model to obtain contextual word embeddings. Additional details on the model architecture are provided in Appendix C.

4 RELATED WORK

Several lines of prior research connect PGMs, deep learning, and meta-learning. We review the most relevant directions and studies below.

Probabilistic graphical models. Various PGMs (e.g., Bayesian Networks (Pearl, 1986), Markov Random Fields (Boykov et al., 1998), latent variable models) have been proposed across diverse domains, such as medical diagnosis (McLachlan et al., 2020), sensing (Diebel and Thrun, 2005), and natural language processing (Blei et al., 2003). However, these models demand careful specification, which can be difficult in heterogeneous or high-dimensional data.

Deep generative architectures such as VAEs (Kingma and Welling, 2013) and Deep Boltzmann Machines (Salakhutdinov and Hinton, 2009; Goan and Fookes, 2020) alleviate some of these challenges by using neural networks to approximate complex conditional distributions. This improves generative capabilities, but often at the expense of the well-defined latent semantics that make PGMs interpretable (Svensson and Pachter, 2019).

Hybrid deep-probabilistic models. Several studies have combined neural networks with the structured reasoning of PGMs. For example, Deep Poisson Factor Analysis (Gan et al., 2015) and Deep Latent Dirichlet Allocation (Cong et al., 2017) replace classical priors and likelihoods with neural parameterizations. However, these approaches are often model-specific and rely heavily on sampling-based inference, which limits their generality and scalability. In contrast, MPM provides a general approach for learning generative models for a broad class of latent variable models.

Structured variational inference. A similar line of work explores combining probabilistic structure with neural inference using variational methods. Structured VAEs (Johnson et al., 2016) augment graphical models with neural components for structured latent representations. However, their framework learns generative mechanisms from a single dataset, which does not capture cross-data structure. Krishnan et al. (2017) integrate VAEs with continuous state-space models, using inference networks to model temporal latent structure. While effective for nonlinear dynamical systems, the approach does not readily extend to more general model classes.

Our method extends structured variational inference in two key aspects: (1) it applies to a broad class of tractable latent variable models, and (2) it explicitly separates global generative structure from dataset-specific variation. We show this approach demonstrates connections to existing models and meta-learning, and is empirically able to discover latent structure within and across complex datasets.

Meta-learning and probabilistic models. Our approach also connects to meta-learning, which is a paradigm that aims to generalize across tasks or datasets (Hospedales et al., 2022). Extensions of meta-learning to probabilistic models include meta-amortized inference (Edwards and Storkey, 2017),

Table 1: On the Tetrominoes dataset, the additive meta-probabilistic model achieves the highest mean ARI score across five runs with different random seeds. In contrast, existing models (i.e. Slot Attention and Slot VAE) exhibit significant variability in performance across runs, while generative baselines are not well suited for the object-centric learning task. Error bars indicate one standard deviation.

Model	ARI (%)
GMM	77.38 ± 0.45
VAE	16.81 ± 9.36
Latent diffusion	14.84 ± 7.41
Slot attention	85.40 ± 13.86
Slot VAE	25.35 ± 34.57
Mixture MPM (Ours)	52.93 ± 3.20
Additive MPM (Ours)	$\mathbf{96.46 \pm 1.89}$

where a dataset-specific context governs the latent space, and probabilistic task models (Nguyen et al., 2021), which combine a VAE with a Gaussian LDA prior for discovering task-themes. While we share a high-level structure with these works, our goal is to provide a more general framework for combining latent variable models with neural architectures. We develop a scalable learning algorithm for this class of models, which provides a probabilistic understanding of the existing Slot Attention architecture.

Other work has linked meta-learning to Bayesian inference. For instance, Grant et al. (2018) show that Model-Agnostic Meta-Learning (Finn et al., 2017) can be interpreted as hierarchical Bayesian inference. While we also adopt a hierarchical Bayesian perspective, our framework differs by explicitly separating global components from local parameters. This enables learning the underlying generative process itself across datasets while still allowing flexible adaptation to dataset-level variation.

5 EXPERIMENTS

We evaluate our proposed method on object-centric learning and sequential text modeling tasks. The experiments demonstrate that MPM (1) discovers shared generative mechanisms, (2) captures dataset-specific latent variables that form semantically meaningful clusters, and (3) identifies high-level latent attributes within each group. All code for our model and experiments are available at: `https://github.com/kzhangm02/mpm`.

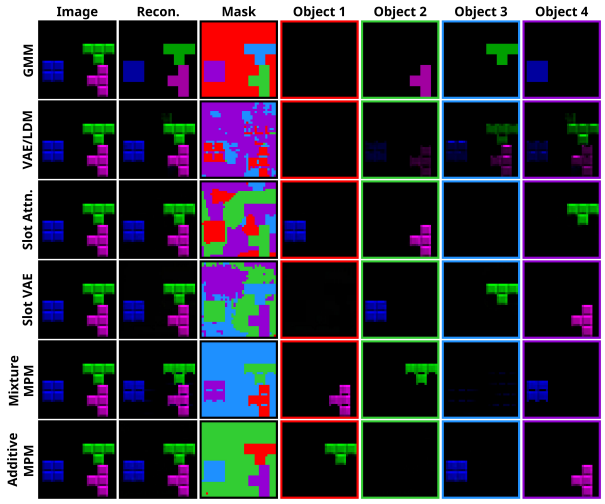


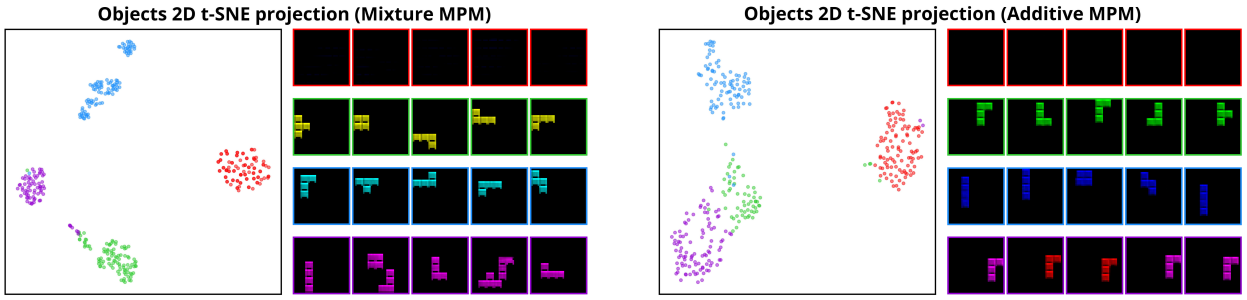
Figure 4: MPM produces high-quality images and more precise object segmentation masks compared to existing architectures. Border colors correspond to the alpha mask colors shown in the third column.

Datasets. For object-centric learning, we use the Tetrominoes dataset (Kabra et al., 2019), which comprises 10,000 images of three non-overlapping shapes. Each shape varies in position, color, and type, selected from a fixed set of tetrominoes. For the text modeling experiments, we use a subset of the AP News corpus (Harman, 1993), consisting of approximately 2,200 news articles from the Associated Press. In both domains, we partition the data into 80% training, 10% validation, and 10% test splits.

5.1 Training and Evaluation

All experiments follow the training procedure in Algorithm 1. In practice, we scale the entropy regularization by a multiplicative factor $\beta < 1$, similar to the β-VAE (Higgins et al., 2016), as this modification yields improved empirical performance. We analyze the effect of the regularization in Appendix D.1.

For object-centric learning, we set $K = 4$ dataset-level clusters corresponding to the three foreground objects and the background, $L = 100$ global object clusters, and $\beta = 0.01$. The model uses the standard convolutional neural network (CNN) architecture. The sequential text modeling experiment uses $K = 5$ dataset-level clusters, $L = 100$ global topic clusters, and $\beta = 0.1$. To obtain contextualized word embeddings for the recognition model g_ϕ, we extract token embeddings from a pretrained BERT model and average the subword representations. Additional training details are provided in Appendix C.



(a) (b)

Figure 5: Our meta-probabilistic model identifies global object clusters that align with latent attributes such as color, position, and shape, under both the mixture (5a) and additive (5b) formulations. The left panel shows the two-dimensional t-SNE embedding, and the right panel displays the five objects with the highest responsibility scores for each cluster. Border colors denote the corresponding cluster assignments.

Following prior object-centric learning work, we evaluate performance using qualitative visualizations and the Adjusted Rand Index (ARI). ARI measures clustering agreement by comparing pairwise assignments. Zero corresponds to random clustering, while one indicates perfect agreement.

For sequential text modeling, we report the maximum and mean UMass topic coherence scores across all topics on the test set. Higher coherence values indicate stronger semantic consistency. We assess interpretability by examining the most representative words for each topic. These terms are selected using the term frequency–inverse document frequency (tf-idf), which identifies distinctive words in a topic.

5.2 Object-centric learning results

Table 1 reports ARI scores for our meta-probabilistic model using both mixture and additive formulations. We compare against classical clustering baselines (GMMs), generative models including a Variational Autoencoder (VAE) and a Latent Diffusion Model (LDM), and established object-centric architectures such as Slot Attention (Locatello et al., 2020) and Slot VAE (Wang et al., 2023).

The additive model achieves the best performance across five training runs. Standard VAEs and LDMs obtain low ARI scores, as their objectives focus on generation rather than structured decomposition. While Slot Attention and object-centric VAEs learn object-level representations, they exhibit significant training instability across random seeds, consistent with Locatello et al. (2020). The mixture variant performs worse than the additive model. Empirically, this is due to a tendency to merge objects with similar color or shape into a single slot.

We visualize the learned structural representations in Figure 4. The GMM baseline groups pixels primarily by color, but fails to capture object-specific structure and fine details. In contrast, the VAE and LDM generate high-quality images yet do not separate individual objects. For brevity, we present their results jointly, as they are visually similar.

Existing object-centric models achieve high-quality and semantically meaningful reconstructions. However, the spatial segmentation is often imprecise, with background pixels assigned to object slots. By comparison, our meta-probabilistic models produce sharp spatial separation between objects and background while maintaining high reconstruction quality and exhibit minimal training instability.

MPM can also discover clusters of objects across images. To visualize these global groupings, we compute responsibility scores $r_{ik\ell}$, which measure the contribution of each global cluster c_ℓ to a given object. For a selected subset of clusters, Figure 5 shows the five objects with the highest responsibility scores, together with their two-dimensional t-SNE embeddings (van der Maaten and Hinton, 2008). The results reveal that the model organizes objects according to latent attributes, such as shape, color, and position, demonstrating that the learned global structure is semantically meaningful. In contrast, most existing object-centric learning architectures cannot identify any global structure across objects.

5.3 Sequential text modeling results

For sequential text modeling, Table 2 reports the UMass coherence score on the test corpus, using the top 10 words from each global topic ranked by tf-idf. Unused topics are excluded from the evaluation.

Table 2: MPM obtains competitive mean and maximum UMass coherence scores across topics in the AP corpus, compared to existing neural topic modeling approaches. We report the mean and standard deviation across five runs with different random seeds.

Model	UMass (mean)	UMass (max)
LDA	-0.753 ± 0.015	-0.215 ± 0.076
NVDM	-1.072 ± 0.028	-0.746 ± 0.065
GSM	-0.500 ± 0.049	-0.224 ± 0.026
NTM	-0.700 ± 0.030	-0.418 ± 0.026
NTMR	-1.071 ± 0.011	-0.704 ± 0.030
WeTe	-0.589 ± 0.015	-0.167 ± 0.115
FASTopic	$\underline{\mathbf{-0.462 \pm 0.019}}$	-0.012 ± 0.012
MPM (Ours)	-0.499 ± 0.059	$\underline{\mathbf{-0.011 \pm 0.008}}$

Because MPM is designed for latent structure discovery across datasets, we compare against neural topic modeling approaches, which typically parameterize probabilistic models with neural networks to infer topic structure. Latent Dirichlet Allocation (LDA) acts as a classical baseline. In our evaluation, we examine the Neural Variational Document Model (NVDM) (Miao et al., 2016), Gaussian Softmax Topic Model (GSM) (Miao et al., 2017), Neural Topic Model and its regularized variant (NTM/NTM-R) (Ding et al., 2018), mixture-based embedding models (WeTe) (Wang et al., 2022), and FASTopic (Wu et al., 2024).

Our method achieves competitive coherence scores relative to prior work. The mean UMass coherence is similar to using FASTopic and GSM, while the maximum score is matched only by FASTopic. We additionally evaluate coherence using Normalized Pointwise Mutual Information in Appendix D.3 and find similar performance. An analysis of scaling behavior is also provided in Appendix D.4.

Figure 6 shows representative words for each topic at both the document and corpus levels. Within individual sentences, topics tend to reflect syntactic structure. For example, the green topic primarily contains punctuation, whereas the purple topic consists of prepositions and articles. We also present five example global topics, which generally exhibit greater semantic coherence, though they occasionally capture syntactic elements such as punctuation.

We emphasize that the aim of our text experiments is not to achieve state-of-the-art performance in topic modeling, but rather demonstrate practical use cases of our general probabilistic modeling methodology, which naturally extends to topic modeling.

Article

Heavy D & The Boyz won a Soul Train Music Award and a NAACP Image Award for their album "Big Tyme." . . .

(a)

Article Topics

group	Expo	,	Dixon	the
was	headline	.	rap	said
The	ground	a	ramp	of
on	freaky	-	barrel	and
an	fooling	"	arena	in

(b)

Corpus Topics

percent	Azerbaijan	.	percent	was
bank	Azerbaijanis	,	Soviet	The
Court	Armenians	"	today	from
billion	Armenian	a	president	on
Tuesday	companies	-	workers	by

(c)

Figure 6: In the test article (6a), document-level topics (6b) primarily capture syntactic structure, whereas corpus-level topics (6c) have more coherent semantic groupings. At each level, the top five words per topic are listed from top to bottom.

6 DISCUSSION

Probabilistic models are often limited by assumptions on the generative process. In this work, we propose a meta-probabilistic modeling method that learns the generative process itself from a collection of related datasets by hierarchically decomposing the generative mechanism into global and dataset-specific parameters. We develop an efficient and scalable training algorithm by deriving a tractable surrogate likelihood bound.

Our experiments show that MPM can effectively combine the expressive modeling capacity of neural networks with the interpretable structure of classical latent variable models. We also demonstrate that the Slot Attention architecture emerges as a special case of our formulation. This perspective allows us to extend the method naturally to tasks such as clustering objects across images based on latent attributes, as well as topic discovery in sequential text datasets.

Limitations. Meta-probabilistic modeling requires multiple related datasets to learn cross-dataset patterns. This dependence is application-specific, and not all problems will fall within this structure. Nonetheless, we provide a principled and natural way to model and learn cross-dataset relationships when they do exist, such as in object-centric modeling.

Acknowledgments

We thank Kartik Ahuja for helpful discussion in forming the foundational idea of this work. YW was supported in part by funding from the Office of Naval Research under grant N00014-23-1-2590, the National Science Foundation under grant No. 2310831, No. 2428059, No. 2435696, No. 2440954, a Michigan Institute for Data Science Propelling Original Data Science (PODS) grant, Two Sigma Investments LP, and LG Management Development Institute AI Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

Edoardo M. Airoldi. Getting started in probabilistic graphical models. *PLoS Computational Biology*, 3, 2007.

Christopher M. Bishop and Michael E. Tipping. A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:281–293, 1998.

David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55:77–84, 2012.

David M. Blei and John D. Lafferty. Dynamic topic models. In *International Conference on Machine Learning*, 2006.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Yuri Boykov, Olga Veksler, and Ramin Zabih. Markov random fields with efficient approximations. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998.

Yulai Cong, Bo Chen, Hongwei Liu, and Mingyuan Zhou. Deep latent dirichlet allocation with topic-layer-adaptive stochastic gradient riemannian mcmc. In *International Conference on Machine Learning*, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

James Diebel and Sebastian Thrun. An application of markov random fields to range sensing. In *Advances in Neural Information Processing Systems*, 2005.

Ran Ding, Ramesh Nallapati, and Bing Xiang. Coherence-aware neural topic modeling. In *Empirical Methods in Natural Language Processing*, 2018.

Andreas Doerr, Christian Daniel, Martin Schiegg, Nguyen-Tuong Duy, Stefan Schaal, Marc Toussaint, and Sebastian Trimpe. Probabilistic recurrent state-space models. In *International Conference on Machine Learning*, 2018.

Harrison Edwards and Amos Storkey. Towards a neural statistician. In *International Conference on Learning Representations*, 2017.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

Zhe Gan, Changyou Chen, Ricardo Henao, David Edwin Carlson, and Lawrence Carin. Scalable deep poisson factor analysis for topic modeling. In *International Conference on Machine Learning*, 2015.

Ethan Goan and Clinton Fookes. Bayesian neural networks: An introduction and survey. *CoRR*, abs/2006.12024, 2020.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *International Conference on Learning Representations*, 2018.

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*, 2019.

Donna Harman. Overview of the first trec conference. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, 1993.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:5149–5169, 2022.

Matthew J. Johnson, David K. Duvenaud, Alex Wiltschko, Ryan P. Adams, and Sandeep R. Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, 2016.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

Biing-Hwang Juang and Lawrence Rabiner. Mixture autoregressive hidden markov models for speech signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33:1404–1413, 1985.

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.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

Rahul Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *AAAI Conference on Artificial Intelligence*, 2017.

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

Hongmei Li, Wenning Hao, Wenyan Gan, and Gang Chen. Survey of probabilistic graphical models. In *10th Web Information System and Application Conference*, 2013.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, 2020.

Scott McLachlan, Kudakwashe Dube, Graham A. Hitman, Norman E. Fenton, and Evangelia Kyrimi. Bayesian networks in healthcare: Distribution by medical condition. *Artificial Intelligence in Medicine*, 107, 2020.

Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. In *International Conference on Machine Learning*, 2016.

Yishu Miao, Edward Grefenstette, and Phil Blunsom. Discovering discrete latent topics with neural variational inference. In *International Conference on Machine Learning*, 2017.

Cuong C. Nguyen, Thanh-Toan Do, and Gustavo Carneiro. Probabilistic task modelling for meta-learning. In *Uncertainty in Artificial Intelligence*, 2021.

Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29:241–288, 1986.

Rishav Pramanik, José-Fabian Villa-Vásquez, and Marco Pedersoli. Masked multi-query slot attention for unsupervised object discovery. In *International Joint Conference on Neural Networks*, 2024.

Lawrence Rabiner and Biing-Hwang Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3:4–16, 1986.

Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, 2009.

Ricardo Silva and Zoubin Ghahramani. The hidden life of latent variables: Bayesian learning with mixed graph models. *Journal of Machine Learning Research*, 10:1187–1238, 2009.

Valentine Svensson and Lior S. Pachter. Interpretable factor models of single-cell rna-seq via variational autoencoders. *Bioinformatics*, 36: 3418–3421, 2019.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305, 2008.

Dongsheng Wang, Dandan Guo, He Zhao, Huangjie Zheng, Korawat Tanwisuth, Bo Chen, and Mingyuan Zhou. Representing mixtures of word embeddings with mixtures of topic embeddings. In *International Conference on Learning Representations*, 2022.

Yanbo Wang, Letao Liu, and Justin Dauwels. Slot-vae: Object-centric scene generation with slot attention. In *International Conference on Machine Learning*, 2023.

Xiaobao Wu, Thong Nguyen, Delvin Ce Zhang, William Yang Wang, and Anh Tuan Luu. Fastopic: Pretrained transformer is a fast, adaptive, stable, and transferable topic model. In *Advances in Neural Information Processing Systems*, 2024.

Ziyi Wu, Jingyu Hu, Wuyue Lu, Igor Gilitschenski, and Animesh Garg. Slotdiffusion: Object-centric generative modeling with diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
   (b) Complete proofs of all theoretical results. [Not Applicable]
   (c) Clear explanations of any assumptions. [Not Applicable]
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials for Meta-probabilistic Modeling

A PROOF OF PROPOSITION 1

Recall the meta-probabilistic loss and surrogate objective:

$$\begin{aligned}
\mathcal{L}^{\mathrm{MPM}}(\lambda^0, \theta, \phi, \eta) &\coloneqq \sum_i \mathcal{L}_i^{\mathrm{ELBO}}(\Lambda_{\phi,\eta}^T, \theta, \eta, q_{\phi,\eta}^T) \\
&\coloneqq \sum_i \left[\log p(\lambda_i \mid \eta) + \sum_j \mathbb{E}_q \left[\log \frac{p_\theta(x_{ij} \mid z_{ij}, \lambda_i) p(z_{ij} \mid \lambda_i)}{q(z_{ij})} \right] \right]. \\
\widehat{\mathcal{L}}^{\mathrm{ELBO}}(\Lambda, \phi, \eta, q) &\coloneqq \sum_i \widehat{\mathcal{L}}_i^{\mathrm{ELBO}}(\Lambda, \phi, \eta, q) \\
&\coloneqq \sum_i \left[\log p(\lambda_i \mid \eta) + \sum_j \mathbb{E}_q \left[\log \frac{\exp\{\psi_\phi(z_{ij} \mid x_{ij}, \lambda_i)\} p(z_{ij} \mid \lambda_i)}{q(z_{ij})} \right] \right],
\end{aligned}$$

where $\psi_\phi(z_{ij} = k \mid x_{ij}, \lambda_i) = -\frac{1}{2}\|\mu_{ik} - g_\phi(x_{ij})\|^2$.

For a fixed i, optimizing $\widehat{\mathcal{L}}_i^{\mathrm{ELBO}}$ with respect to q is equivalent to maximizing

$$\sum_j \mathbb{E}_q \left[\log \frac{\exp\left(-\frac{1}{2}\|\mu_{iz_{ij}} - g_\phi(x_{ij})\|^2\right)}{q(z_{ij})} \right],$$

which is the negative KL divergence between q and the unnormalized distribution $\exp\left(-\frac{1}{2}\|\mu_{iz_{ij}} - g_\phi(x_{ij})\|^2\right)$. Thus, the optimal q satisfies

$$q(z_{ij} = k) \propto \exp\left(-\frac{1}{2}\|\mu_{ik} - g_\phi(x_{ij})\|^2\right).$$

We find the maximizing of Λ by setting the gradient to zero. For a fixed μ_{ik},

$$\begin{aligned}
\nabla_{\mu_{ik}} \widehat{\mathcal{L}}^{\mathrm{ELBO}}(\Lambda, \phi, \eta, q) &= \nabla_{\mu_{ik}} \left[\log p(\mu_{ik} \mid \eta) + \sum_j \mathbb{E}_q[\psi_\phi(z_{ij} \mid x_{ij}, \lambda_i)] \right] \\
&= \nabla_{\mu_{ik}} \left[\log \left(\sum_{\ell=1}^L \exp\left(-\frac{1}{2}\|\mu_{ik} - \nu_\ell\|^2\right) \right) - \frac{1}{2} \sum_{j=1}^{N_i} \sum_{k=1}^K q(z_{ij} = k) \cdot \|\mu_{ik} - g_\phi(x_{ij})\|^2 \right] \\
&= -\left[\sum_{\ell=1}^L r_{ik\ell}(\mu_{ik} - \nu_\ell) + \sum_{j=1}^{N_i} s_{ijk}(\mu_{ik} - g_\phi(x_{ij})) \right],
\end{aligned}$$

where

$$r_{ik\ell} = \frac{\exp\left(-\frac{1}{2}\|\mu_{ik} - \nu_\ell\|^2\right)}{\sum_{\tilde{\ell}} \exp\left(-\frac{1}{2}\|\mu_{ik} - \nu_{\tilde{\ell}}\|^2\right)}, \quad s_{ijk} = \frac{\exp\left(-\frac{1}{2}\|\mu_{ik} - g_\phi(x_{ij})\|^2\right)}{\sum_{\tilde{k}} \exp\left(-\frac{1}{2}\|\mu_{i\tilde{k}} - g_\phi(x_{ij})\|^2\right)}.$$

Setting the gradient to zero yields the update for μ_{ik}:

$$\mu_{ik} = \frac{\sum_\ell r_{ik\ell}\nu_\ell + \sum_j s_{ijk} g_\phi(x_{ij})}{\sum_\ell r_{ik\ell} + \sum_j s_{ijk}}.$$

This provides the update steps used in the meta-probabilistic inference procedure in our two case studies. □

Algorithm 2 Training procedure for Slot Attention

Require: Dataset $\mathcal{D} = \{x_i\}_{i=1}^n$, inner optimization steps T, learning rate α
Output: Parameters $\theta, \phi, \mu, \sigma$

Initialize $\vartheta = \{\theta, \phi, \mu, \sigma\}$
while not converged **do**
  Sample $x \in \mathsf{Uniform}(\mathcal{D})$
  Sample $\varepsilon_k \sim \mathcal{N}(0, I_d)$
  $z \leftarrow \mathsf{LayerNorm}(g_\phi(x))$
  $s^{(0)} \leftarrow \mu + \sigma \cdot \varepsilon$
  for $t = 0$ **to** $T - 1$ **do**
    $s^{(t)}_{\text{prev}} \leftarrow s^{(t)}$
    $s^{(t)} \leftarrow \mathsf{LayerNorm}(s^{(t)})$
    $A^{(t)} \leftarrow \mathsf{Softmax}\left(\frac{(s^{(t)}W_q)(zW_k)^\top}{\sqrt{d}}, \mathsf{axis} = \mathsf{slots}\right)$
    $u^{(t)} \leftarrow \mathsf{WeightedMean}(\mathsf{weights} = A^{(t)}, \mathsf{values} = zW_v)$
    $s^{(t+1)} \leftarrow \mathsf{GRU}(\mathsf{state} = s^{(t)}_{\text{prev}}, \mathsf{update} = u^{(t)})$
    $s^{(t+1)} \leftarrow s^{(t+1)} + \mathsf{MLP}(\mathsf{LayerNorm}(s^{(t+1)}))$
  end for
  $\mathcal{L} \leftarrow \|x - f_\theta(s^{(T)})\|^2$
  $\vartheta \leftarrow \mathtt{SGD}(\vartheta, \nabla_\vartheta \mathcal{L}, \alpha)$
end while
return $\theta, \phi, \mu, \sigma$

B SLOT ATTENTION DETAILS

We provide an overview of the Slot Attention architecture for object-centric learning. Let $\mathcal{X} \subseteq \mathbb{R}^{H \times W \times 3}$ denote the image space, and let $g_\phi : \mathcal{X} \to \mathcal{Z}$ be a deterministic encoder that maps an image to a latent representation, where $\mathcal{Z} \subseteq \mathbb{R}^{N \times d}$ consists of N feature vectors in $\mathbb{R}^d$. The *Slot Attention module* transforms these features into a set of K slots, each represented in the slot space $\mathcal{S} \subseteq \mathbb{R}^d$. A deterministic decoder $f_\theta : \mathcal{S}^K \to \mathcal{X}$ then maps the slots back to the image space. Typically, the encoder and decoder are implemented using CNNs with spatial broadcasting and positional encodings. Alternative architectures based on pre-trained Vision Transformers or latent diffusion models have also been proposed (Wu et al., 2023; Pramanik et al., 2024).

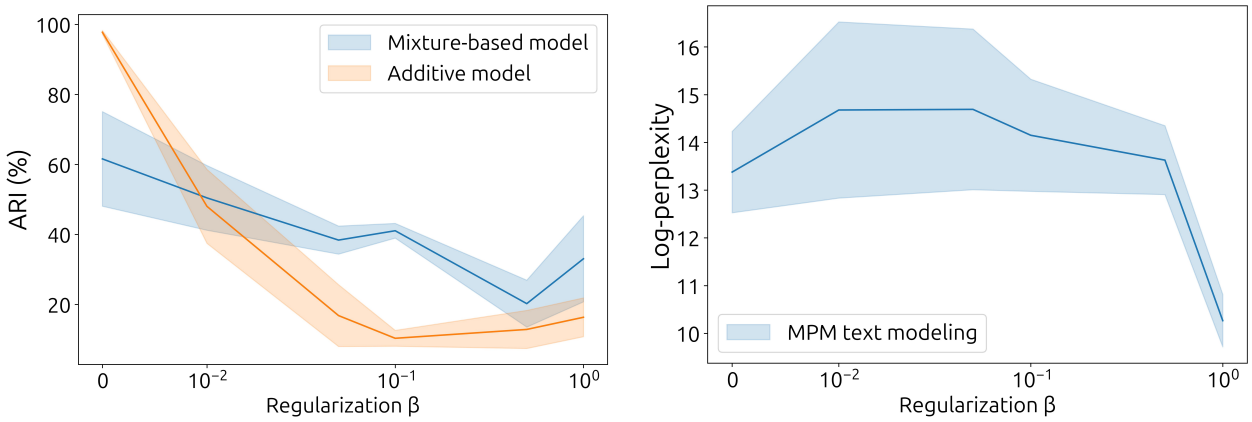
For each image, the Slot Attention module receives a set of N feature vectors in $\mathbb{R}^d$ and uses an iterative attention mechanism to refine a set of K slots. To formalize this process, let $z \in \mathcal{Z}$ denote the input set of features, and let $s^{(t)} \in \mathcal{S}^K \subseteq \mathbb{R}^{K \times d}$ denote the slot representations at iteration t. Let $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$ denote the query, key, and value projection matrices, respectively.

At iteration t, the attention weights $A^{(t)}$ and slot update $u^{(t)}$ are computed as

$$u^{(t)} = B^{(t)}(zW_v), \quad \text{where} \quad B_{ij} \coloneqq \frac{A_{ij}}{\sum_k A_{ik}}, \quad A^{(t)} = \mathsf{Softmax}\left(\frac{(s^{(t)}W_q)(zW_k)^\top}{\sqrt{d}}\right) \in \mathbb{R}^{K \times N}.$$

The slots are updated according to $s^{(t+1)} = \mathsf{SlotUpdate}(s^{(t)}, u^{(t)})$, which consists of a Gated Recurrent Unit (GRU) followed by a multilayer perceptron (MLP) with a residual connection. The initial slots $s^{(0)}$ are randomly sampled from a learned Gaussian distribution. This iterative refinement is performed for T rounds to produce the final slots $s^{(T)}$.

We next detail the additive slot decoder used to reconstruct the image from the set of slots. The decoder maps each slot to a masked image space $\mathcal{X}_m \subseteq \mathbb{R}^{H \times W \times (C+1)}$, where the image consists of C channels together with an additional unnormalized alpha channel. Let $h_\theta : \mathcal{S} \to \mathcal{X}_m$ denote a per-slot decoder that maps each slot to an object in the image.



(a) (b)

Figure 7: For the object-centric learning meta-probabilistic models, ARI decreases sharply as the regularization strength increases (Figure 7a). In contrast, the log-perplexity of our model in the text experiments remains mostly stable across values of β (Figure 7b). For each value of β, we run five trials with different random initializations and training splits and report the mean and standard deviation.

Given slots $s = \{s_1, \ldots, s_K\}$, let $\alpha_1, \ldots, \alpha_K$ denote the normalized alpha masks obtained by applying a softmax across slots. Then,

$$f_\theta(s) = \sum_{i=1}^{K} \alpha_i h_\theta(s_i).$$

Hence, we refer to this architecture as an *additive decoder*. The full training procedure for Slot Attention is outlined in Algorithm 2. For simplicity, we omit the batch dimension, but the algorithm can be easily modified to accommodate minibatches. We refer the reader to Locatello et al. (2020) for additional details.

At this point, the algorithmic similarities between MPM and Slot Attention should begin to emerge. Like MPM, Slot Attention uses a bi-level scheme in which the slots are iteratively refined for each instance during training. We show that when viewed as clustering in the latent space, the model is effectively maximizing a bound on the data likelihood, thus providing a principled explanation for its effectiveness.

C TRAINING DETAILS

For object-centric image modeling, we adopt a convolutional neural network (CNN) architecture for both the generative model and the recognition network, following an encoder–decoder style design. Models are optimized with Adam using an initial learning rate of 4×10^{-4} and step-based learning rate decay, which we find produces stable convergence across runs. We train for 1,000 epochs, which requires approximately one hour for our model, and twice as long for Slot Attention.

The sequential text model is parameterized as a multinomial distribution over tokens, conditioned on a topic embedding, produced by a three-layer MLP. The recognition network uses a frozen pre-trained BERT model (Devlin et al., 2019), followed by a trainable two-layer MLP to generate contextual embeddings for each token. Word-level embeddings are obtained by averaging subword token embeddings, and articles are truncated to 512 tokens to align with BERT's maximum input length. We use the Adam optimizer with an initial learning rate of 1×10^{-5} and step-based learning rate decay, over 200 epochs. The training requires roughly 1.5 hours.

All experiments were performed on a single NVIDIA RTX 5070 GPU with 16GB memory. We tune learning rates via grid search over $\{1 \times 10^{-5}, 4 \times 10^{-5}, 1 \times 10^{-4}, 4 \times 10^{-4}, 1 \times 10^{-3}\}$. The hyperparameter β is selected to be as large as possible from $\{0.01, 0.05, 0.1, 0.5, 1.0\}$ without significantly degrading reconstruction quality.

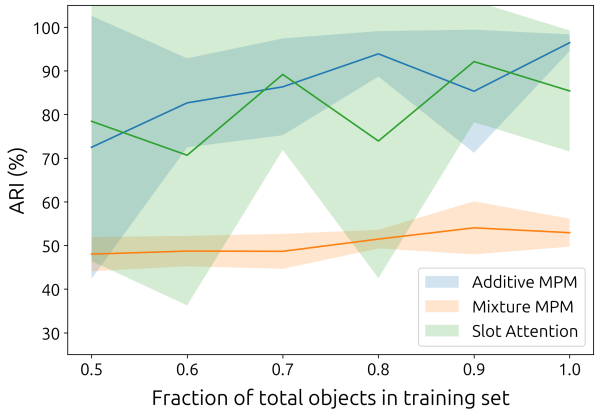


Figure 8: Object-centric learning performance gradually declines as task diversity decreases, as measured by the fraction of objects seen in the training data.

Table 3: MPM achieves mean NPMI scores comparable to the baselines, while attaining competitive maximum NPMI values. We report the mean and standard deviation across five runs with different random seeds and initializations.

Model	NPMI (mean)	NPMI (max)
LDA	0.165 ± 0.005	0.599 ± 0.145
NVDM	0.070 ± 0.002	0.157 ± 0.011
GSM	0.145 ± 0.003	0.424 ± 0.041
NTM	0.217 ± 0.011	0.465 ± 0.045
NTMR	0.072 ± 0.001	0.143 ± 0.016
WeTe	$\underline{\mathbf{0.230 \pm 0.003}}$	0.617 ± 0.028
FASTopic	0.207 ± 0.004	$\underline{\mathbf{0.724 \pm 0.067}}$
MPM (Ours)	0.138 ± 0.022	0.624 ± 0.181

D ADDITIONAL EXPERIMENTS

D.1 Effect of the regularization parameter

We examine how the regularization parameter β influences model performance. Specifically, we vary β logarithmically from 10^{-2} to 1, and additionally evaluate the unregularized case $\beta = 0$. Figure 7 reports the ARI for the mixture and additive decoder models, along with the log-perplexity of the sequential text model. In the object-centric learning setting, performance degrades sharply as β increases. This behavior arises because stronger regularization encourages the posterior distribution to become more uniform, which suppresses the underlying clustering structure. In contrast, for the sequential text modeling task, performance remains relatively stable across the range of β, with a modest improvement observed at $\beta = 1$.

D.2 Effect of task diversity on object-centric learning

Meta-learning algorithms typically rely on the assumption that the underlying tasks exhibit sufficient diversity. As with other meta-learning approaches, our model also depends on the presence of meaningful cross-dataset patterns in order to learn transferable global structure.

To study how our model behaves under varying levels of task diversity, we use the number of distinct objects present in the training data as a proxy for task diversity. Each object corresponds to a tetromino characterized by a specific color, shape, and orientation. We construct restricted training datasets by sampling a random fraction of the full set of possible objects and retaining only images that contain objects from this subset. The fraction of allowed objects is varied over $\{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, where including all objects corresponds to the original experimental setting. Because restricting the object set reduces the number of available training images, we apply rotation and flip augmentations to keep the training set size consistent across settings.

For each level of task diversity, we evaluate across five random object restrictions and training seeds. Figure 8 reports the mean and standard deviation of the ARI scores. As expected, all models exhibit a decline in ARI as the number of observed objects in the training data decreases, reflecting the reduced task diversity. The additive MPM and Slot Attention models show a gradual performance decline due to their compositional behavior, while the mixture model remains relatively stable across different levels of task diversity.

We note that restricting object types introduces a distribution shift between the training and test data, since the test set may contain objects that are absent from the training images. Nonetheless, the number of allowed object types serves as a useful proxy for task diversity: when fewer object types are available, the resulting training images become more similar to one another, effectively reducing the diversity of tasks.

Table 4: When using BERT as the contextual word embedding model in the text experiments, our meta-probabilistic models incur a similar runtime cost to fine-tuning BERT, as the inner optimization is lightweight. However, memory usage is higher than standard fine-tuning due to the additional optimization steps.

Max sequence length	Time (MPM)	Time (BERT-FT)	GPU (MPM)	GPU (BERT-FT)
64	0.3 hr	0.2 hr	3 GB	2 GB
128	0.4 hr	0.3 hr	5 GB	3 GB
256	0.6 hr	0.5 hr	10 GB	4 GB
512	1.2 hr	1.0 hr	16 GB	7 GB

D.3 NPMI topic coherence results

In addition to evaluating our sequential text model using the UMass coherence score, we also report the Normalized Pointwise Mutual Information (NPMI) in Table 3. NPMI measures the co-occurrence frequency between top words within a topic and normalizes using log-probability. Our meta-probabilistic model achieves a maximum NPMI coherence comparable to existing neural topic modeling approaches, including WeTe and FasTOPIC. However, the mean NPMI score is lower, indicating that while some topics exhibit strong coherence, the overall performance remains within the mid-range of the evaluated models.

D.4 Scaling behavior of MPM text model

We conduct a study based on the text experiment to directly examine how the runtime and memory of our meta-probabilistic model scale with data dimensionality. Specifically, we vary the maximum input sequence length and report the resulting computational cost in Table 4 for our meta-probabilistic model (MPM) and for fine-tuning BERT (BERT-FT) as a baseline. Training times for MPM roughly align with BERT-FT, while there is an increase in GPU memory usage due to the additional computation graph introduced by the inner optimization.

In general, we find that the scaling behavior of our models largely depends on the choice of the underlying models f_θ and g_ϕ. Empirically, we find that the runtime is dominated by these components, since the bilevel optimization overhead is relatively small. On the other hand, memory usage can exhibit more noticeable increases because intermediate activations in the iterative procedure are retained.

E EXPERIMENTAL ASSETS

All code for our meta-probabilistic models are publicly available.[1] In the experiments, we use the publicly available Tetrominoes (Kabra et al., 2019) dataset, distributed under the Apache License, and AP News corpus (Harman, 1993) dataset. For the text experiments, we follow the original LDA work (Blei et al., 2003) by using a subset of the AP News corpus, which is available under the GNU Lesser General Public License.[2] For comparison with Slot Attention, we use our own implementation based on a publicly available version.[3]

[1] `https://github.com/kzhangm02/mpm`

[2] `https://github.com/blei-lab/lda-c`

[3] `https://github.com/evelinehong/slot-attention-pytorch`