Содержание

1 Введение **3**

2 Основные определения и постановка задачи **5**

3 Обзор современного состояния проблемы **6**

3.1 Модули выделения слотовых представлений 6

3.2 Объектно-центричные модели 9

4 Метод **19**

4.1 Энкодер 19

4.2 Инициализатор слотовых представлений 20

4.3 Модуль выделения слотовых представлений 20

4.4 Декодер 22

5 Эксперименты **24**

5.1 Наборы данных 24

5.2 Детали обучения 24

5.3 Результаты 25

5.4 Анализ эффективности модификаций 30

6 Заключение **32**

Аннотация

Восприятие и познание человеком окружающей среды опираются на способность разлагать сложные сцены на объектно-центричные (объектно-ориентированные) представления. Недавние достижения в области сегментации объектов без учителя позволили получать структурированные и интерпретируемые представления, что улучшает результаты в широком спектре задач компьютерного зрения. Несмотря на то, что объектно-центричное обучение показало многообещающие результаты, существующие методы зачастую испытывают трудности с обобщением в разнообразных условиях и обычно ограничиваются либо синтетическими, либо реальными сценариями. Slot Attention является одной из доминирующих архитектур для обучения объектно-центричным представлениям. Альтернативный подход, основанный на моделировании смесей гауссовских расределений — известный как Slot Mixture Module (SMM), — предлагает концептуальные преимущества, заменяя механизм перекрёстного внимания вероятностными компонентами смеси. Однако SMM демонстрирует нестабильную сходимость и ограниченную эффективность на реальных данных. В данной работе предлагается новый механизм обучения для slot-based моделей, который значительно повышает их устойчивость и адаптивность. Наш метод использует иерархическую агрегацию ключевых признаков с энкодера в сочетании с итеративной процедурой уточнения на каждом шаге обновления слотов. Веса псевдо-внимания и параметры компонентов смеси моделируются вероятностно и уточняются после обновления ключевых параметров смеси. Мы демонстрируем заметное преимущество нашего метода относительно лучших объектно-центричных решений в задаче сегментации без учителя как на синтетических, так и на реальных данных.

1 Введение

Суть человеческого восприятия и мышления заключается в разложении сложных сцен на объектно-ориентированные абстракции. Для более эффективного понимания и использования графических представлений в компьютерном зрении решающее значение имеет неконтролируемое (без учителя) объектно-центричное обучение. В последние годы интерес к объектно-центричным представлениям в значительной степени возрос, поскольку объектно-центричное обучение потенциально повышает надёжность, эффективность, генерализацию моделей и интерпретируемость получаемых результатов в различных областях, таких как обучение с подкреплением, структурированное представление и генерация сцен, моделирование рассуждений о визуальных сценах, специфичные объектно-центричные визуальные задачи и алгоритмы визуального планирования.

В объектно-ориентированном обучении происходит декомпозиция сцены, в ходе которой модель учится разбивать изображение на несколько компонентов без использования каких-либо размеченных человеком данных, т.е. без учителя. Автоматическая сегментация объектов на сцене и формирование структурированного латентного пространства представлений могут быть реализованы различными способами: дополнение выделяемых признаков информацией о принадлежности к некоторой группе (классу объекта), использование представления тензорного произведения, привлечение идей из динамики аттракторов и т.д. Широкое распространение в объектно-центричном обучении получили модели, основанные на слотах, каждый из которых является абстрактным представлением определённого объекта на сцене.

Несмотря на актуальность исследований и большое количество работ в обсуждаемой области, одним из серьёзнейших вызовов в данный момент является масштабирование методов объектно-центричного обучения на сложные данные реального мира, т.к. большинство моделей показывают хорошие результаты на синтетических данных, но терпят неудачу на реальных. Целью работы является создание универсального и конкурентоспособного метода, достигающего отличных результатов на синтетических наборах и при этом имеющего приемлемое качество на сложных данных реального мира. Для достижения поставленной цели были сформулированы и решены следующие задачи:

1. Обзор работ на тему различных подходов к реконструкции сцен и обнаружению объектов на ней.

2. Разработка алгоритма для стабилизации обучения модуля выделения слотовых представлений.

3. Интеграция наработок в традиционную архитектуру объектно-ориентированных моделей.

4. Проведение экспериментов и анализ полученных результатов.

Во 2-м разделе вводятся основные определения и описывается общая постановка задачи. В 3-м разделе приводится обзор работ и текущего состояния обсуждаемой области. В 4-м разделе описываются предложенный метод и предлагаемый алгоритм стабилизации обучения слотовых представлений, который также повышает и их качество, а также демонстрируются результаты сравнения с базовым решением. В 5-м разделе заключена экспериментальная часть работы, где описываются используемые наборы данных и демонстрируется превосходство предлагаемого метода над базовыми моделями. В заключении подводятся итоги проделанной работы и обсуждаются направления дальнейшего исследования.

2 Основные определения и постановка задачи

Рассмотрим следующие объекты и понятия. Изначально заданы:

- конечное множество объектов $\Omega = \{\omega_1, \omega_2, \ldots, \omega_T\}$, где $|\Omega| = T$;
- пространство признаковых описаний $\mathbb{X} \subset \mathbb{R}^{H \times W \times 3}$, где каждый элемент $\mathbf{x}_t = \varphi(\omega_t)$ представляет собой признаковое описание объекта ω_t;
- функция-отображение $\varphi : \Omega \to \mathbb{X}$, сопоставляющая каждому объекту ω_t соответствующее признаковое описание $\mathbf{x}_t \in \mathbb{X}$.

Пусть также заданы два отображения:

$$f_{\mathbf{enc}} : \mathbb{X} \to \mathbb{X}^*, \qquad f_{\mathbf{dec}} : \mathbb{X}^* \to \hat{\mathbb{X}}, \tag{2.1}$$

где $f_{\mathbf{enc}}$ — это кодирующая функция, переводящая признаковые описания из исходного пространства $\mathbb{X}$ в компактное латентное пространство $\mathbb{X}^* \subset \mathbb{R}^d$, а $f_{\mathbf{dec}}$ — декодирующая функция, восстанавливающая признаковые описания из латентного представления, так что $\hat{\mathbb{X}} \subset \mathbb{R}^{H \times W \times 3}$ представляет собой реконструированные признаковые описания.

На промежуточном этапе (в узком месте модели) могут решаться дополнительные задачи, такие, например, как классификация, кластеризация, позволяющих разделять различные сущности внутри объекта. Решению этих задач способствует формирование множества слотовых представлений $\mathcal{S}$, которые будут использоваться далее для анализа и интерпретации данных.

Определение 2.1. *Пусть $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$ — множество классов объектов, $A(c_i)$ — множество атрибутов класса $c_i \in \mathcal{C}$, $\mathrm{Dom}(c.a_j)$ — множество допустимых значений атрибута $c.a_j$, а $O = \{o_1, o_2, \ldots, o_l\}$ — множество объектов среды, где $o_n \in c_i$ для некоторого i. Тогда пусть $s_n \in \mathbb{R}^N$ — вектор, представляющий все атрибуты объекта o_n. Множество $\mathcal{S} = \{s_1, s_2, \ldots, s_l\}$ называется множеством* ***слотовых представлений****.*

Целью является минимизация расхождения между исходными признаками $\mathbf{x}_t$ и их реконструкцией $\hat{\mathbf{x}}_t = f_{\mathbf{dec}}(f_{\mathbf{enc}}(\mathbf{x}_t))$. Таким образом, задача относится к типу задач регрессии. Для удобства дальнейших рассуждений обозначим $\mathcal{P} = f_{\mathbf{dec}}(\mathbb{X}^*)$, то есть $\mathcal{P} = \{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_T\}$ — множество реконструированных признаковых описаний. Тогда задача оптимизации в простейшем случае сводится к минимизации функционала:

$$\mathcal{Q} = \min_{\hat{\mathbf{x}}_t \in \mathcal{P}} \sum_{t=1}^{T} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2. \tag{2.2}$$

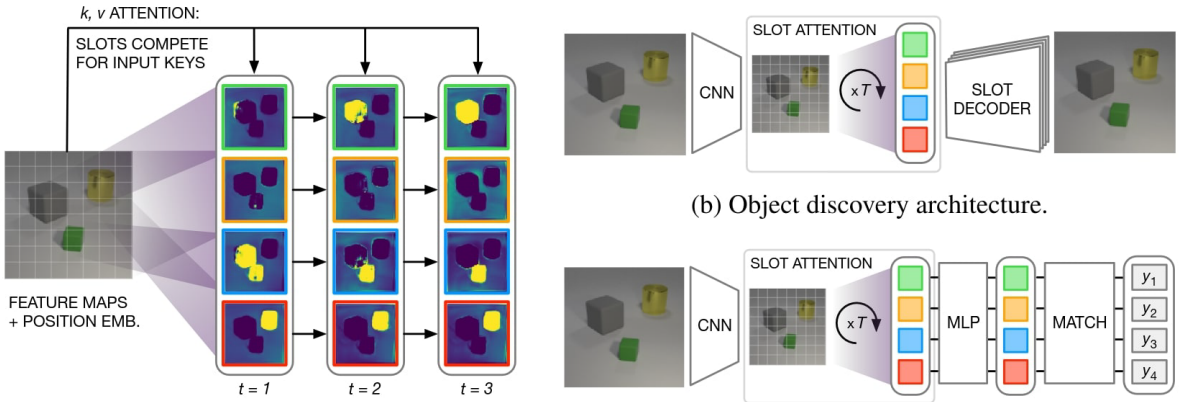
3 Обзор современного состояния проблемы

3.1 Модули выделения слотовых представлений

Фундаментальной проблемой объектно-ориентированного обучения моделей является эффективное формирование слотовых представлений. Качество этих представлений непосредственно влияет на эффективность решения различных задач и общее качество алгоритмов обучения моделей.

Slot Attention. Большинство современных объектно-ориентированных архитектур опираются на модуль Slot Attention [25], который стал основополагающим компонентом в этой области. Slot Attention использует нормализованный по слотам механизм перекрестного внимания [10], который позволяет итеративно назначать слоты различным областям изображения с помощью конкурентного обучения. Его эффективность и масштабируемость сделали его стандартным строительным блоком во многих современных объектно-ориентированных платформах. Схема модуля представлена на рисунке 1, а псевдокод – в алгоритме 3.1.

Slot Mixture. В 2024 году был предложен модуль Slot Mixture [42], схема которого представлена на рисунке 2, а псевдокод – в алгоритме 3.2. Он представляет собой новый подход к объектно-ориентированному обучению представлению, основанный на обобщении модели смесей гауссовских распределений (GMM). В отличие от Slot Attention, который представляет слоты исключительно в виде центроидов кластеров, SMM дополнительно включает информацию о расстояниях между кластерами и назначенных векторах признаков. Это позволяет создавать более выразительные представления фрагментов, что приводит к повышению производительности в таких задачах, как прогнозирование свойств нескольких объектов, реконструкция изображений и обнаружение объектов, особенно при жестких пороговых значениях оценки. Экспериментальные результаты показывают, что SMM достигает самых высоких результатов в задаче прогнозирования заданных свойств набора данных CLEVR. Более того, он показывает превосходные результаты в сложных синтетических наборах данных, таких как ClevrTex, и в реальных данных, таких как COCO. Метод объединяет ключевые этапы классического алгоритма максимизации математического ожидания (EM) с обучаемыми обновлениями нейронной сети, обеспечивая более точное разделение объектов и устойчивость к изменениям входных данных.



(a) Slot Attention module.

(b) Object discovery architecture.

(c) Set prediction architecture.

Рис. 1: Архитектура Slot Attention: a) архитектура модуля, b) архитектура для задачи обнаружения объектов, c) архитектура для предсказания классов объектов.

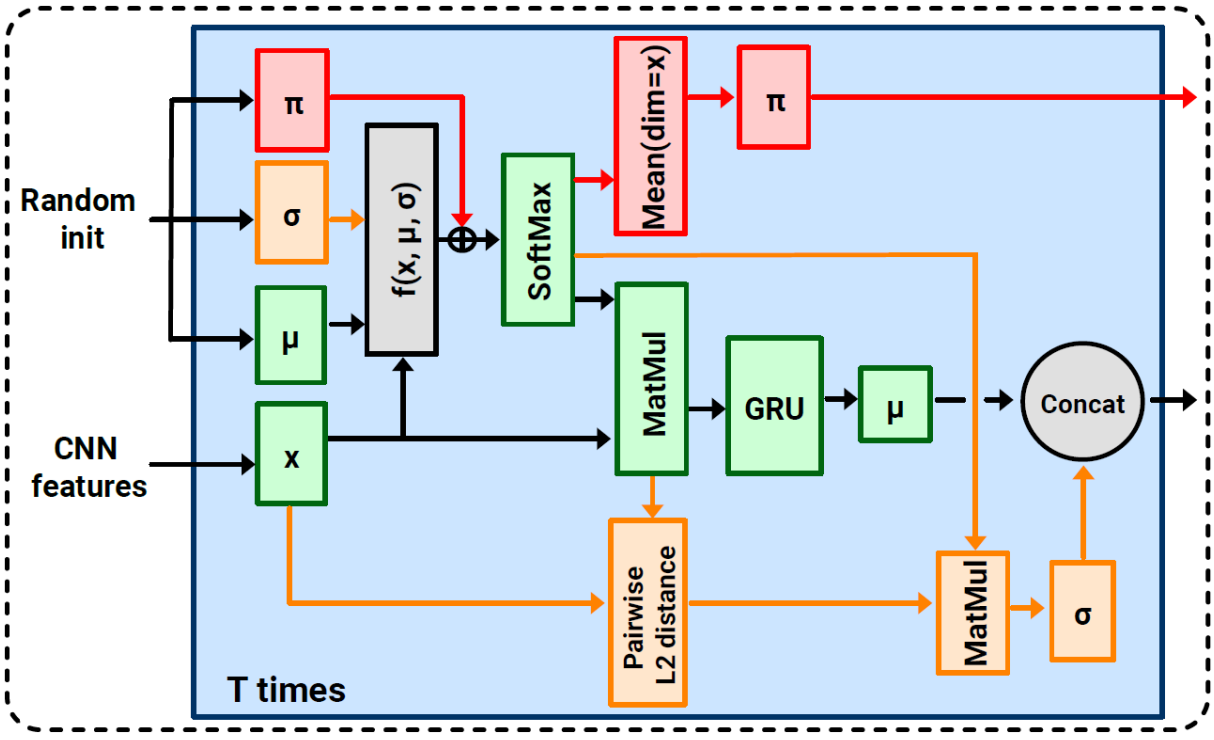


Рис. 2: Архитектура Slot Mixture.

Алгоритм 3.1 Slot Attention модуль. На вход подаётся набор из N векторов размерности $D_{\texttt{inputs}}$, которые назначаются на набор из K слотов размерности $D_{\texttt{slots}}$. Слоты инициализируются независимо друг от друга путём выборки из нормального распределения с общими обучаемыми параметрами $\mu \in \mathbb{R}^{D_{\texttt{slots}}}$ и $\sigma \in \mathbb{R}^{D_{\texttt{slots}}}$. Количество итераций уточнения слотовых представлений $T = 3$.

Вход: $\texttt{inputs} \in \mathbb{R}^{N \times D_{\texttt{inputs}}}$, $\texttt{slots} \sim \mathcal{N}(\mu,\ \mathrm{diag}(\sigma)) \in \mathbb{R}^{K \times D_{\texttt{slots}}}$

Параметры слоёв: $k,\ q,\ v$: линейные проекции для механизма внимания; `GRU`; `MLP`; `LayerNorm` (x3)

$\texttt{inputs} = \texttt{LayerNorm}(\texttt{inputs})$

для $t = 0 \dots T$

$\texttt{slots_prev} = \texttt{slots}$

$\texttt{slots} = \texttt{LayerNorm}(\texttt{slots})$

$\texttt{attn} = \texttt{softmax}(\frac{1}{\sqrt{D}} k(\texttt{inputs}) \cdot q(\texttt{slots})^T,\ \texttt{axis='slots'})$

$\texttt{updates} = \texttt{WeightedMean}(\texttt{weights=attn} + \epsilon,\ \texttt{values=}v(\texttt{inputs}))$

$\texttt{slots} = \texttt{GRU}(\texttt{state=slots_prev},\ \texttt{inputs=updates})$

$\texttt{slots} \mathrel{+}= \texttt{MLP}(\texttt{LayerNorm}(\texttt{slots}))$

return `slots`

Probabilistic Slot Attention. Еще одна недавняя работа – Probabilistic Slot Attention (PSA) [41] – представляет собой вероятностную структуру для объектно-ориентированного обучения, основанную на идее введения смеси представлений. Схема модуля представлена на рисунке 3, а псевдокод – в алгоритме 3.3. PSA создает локальные гауссовы смеси для каждой входной выборки и использует процедуру, подобную EM, для оценки параметров слотов, где каждый слот моделируется как выборка из многомерного нормального распределения. Авторы теоретически обосновали, что этот подход обеспечивает неконтролируемую идентифицируемость объектно-ориентированных представлений вплоть до аффинных преобразований и перестановки ячеек. Эта теоретическая гарантия повышает надежность и согласованность модели при различных начальных условиях.

Основное различие между модулями PSA и SMM заключается в их подходах к моделированию. PSA фокусируется на вероятностном моделировании с помощью GMM для каждого экземпляра, используя агрегированные апостериорные распределения в качестве оптимальных исходных данных. В отличие от этого, SMM рас-

Алгоритм 3.2 The Slot Mixture модуль. $\boldsymbol{\pi}$ инициализируются из равномерного категориального распределения, $\boldsymbol{\mu}$ и $\boldsymbol{\Sigma}_{diag}$ инициализируются из нормального распределения с обучаемыми параметрами.

Вход: $\boldsymbol{x} \in \mathbb{R}^{N \times D}$ — выпрямленные в вектор карты признаков с добавленными позиционными эмбеддингами; $\boldsymbol{\mu}, \boldsymbol{\Sigma}_{diag} \in \mathbb{R}^{K \times D}, \boldsymbol{\pi} \in \mathbb{R}^{K}$ — параметры смеси.
Выход: slots $\in \mathbb{R}^{N \times 2D}$
$\boldsymbol{x} = \text{MLP}(\text{LayerNorm}(\boldsymbol{x}))$
для $t = 0...T$
  logits $= f_\theta(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}_{diag})$
  $\gamma = \text{SoftMax}(\text{logits} + \log \boldsymbol{\pi}, \text{dim=1})$
  $\boldsymbol{\pi} = \gamma.\text{mean}(\text{dim=0})$
  $\boldsymbol{\mu}^* = \text{WeightedMean}(\text{weights=}\gamma, \text{values=}\boldsymbol{x})$
  $\boldsymbol{\mu} = \text{GRU}(\text{input=}\boldsymbol{\mu}^*, \text{hidden=}\boldsymbol{\mu})$
  $\boldsymbol{\mu} \mathrel{+}= \text{MLP}(\text{LayerNorm}(\boldsymbol{\mu}))$
  $\boldsymbol{\Sigma}_{diag} = \text{WeightedMean}(\text{weights=}\gamma, \text{values=}(\boldsymbol{x} - \boldsymbol{\mu})^2)$
slots $= \text{concat}([\boldsymbol{\mu}, \boldsymbol{\Sigma}_{diag}], \text{dim=1})$
return slots

ширяет классический Slot Attention, явно интегрируя обобщенную структуру GMM, используя итеративные обновления в стиле EM для создания более содержательных слотовых представлений. В то время как SMM кодирует слоты как объединения векторов средних и элементов диагональной ковариационной матрицы для повышения способности различать похожие объекты, PSA подчеркивает статистическую идентифицируемость и стабильность представлений благодаря вероятностной формулировке.

3.2 Объектно-центричные модели

SLATE. SLot Attention TransformEr (SLATE) относится к unsupervised (без учителя) подходам в объектно-ориентированном обучении моделей. SLATE – это аналог DALL-E [33], но генерирующий изображения, не нуждаясь в текстовых «подсказках».

Основной вклад SLATE в область обучения объектно-ориентированным представлениям (ООП) можно описать в нескольких пунктах:

1. Первая объектно-ориентированная модель, основанная на Transformer-декодере.

2. Демонстрация значительного улучшения систематической генерализации моделей ООП.

3. Простота модели.

SLATE обеспечивает глобальную семантическую согласованность в zero-shot подходе без текстовой «подсказки», а только по изображениям путём изучения слотов объектов. В качестве декодера используется Transformer [10], заменяя текстовую «подсказку» на слотовую, взятую из библиотеки повторно используемых визуальных концепций. SLATE стремится достичь систематического обобщения в задаче zero-shot генерации изображений. Архитектура SLATE схематично представлена на рисунке 4.

Алгоритм 3.3 Probabilistic Slot Attention

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

Discrete Variational Autoencoder (DVAE). Сначала изображение $\mathbf{x}_t$ разбивается на фрагменты (участки) размера $P \times P$, в результате чего получаем L фрагментов, где $L = \frac{HW}{P^2}$, где H и W – высота и ширина изображения. Хотим получать логарифмы вероятностей релаксированного категориального распределения [12, 9]:

$$U_{t,l} \sim \text{Uniform}(0,1),\ \mathbf{g}_{t,l} = \log \mathbf{x}^*_{t,l} - \log(-\log(U_{t,l} + \epsilon) + \epsilon),\ \mathbf{z}^{\text{soft}}_{t,l} = \underset{L}{\text{softmax}}(\mathbf{g}_{t,l}), \quad (3.1)$$

где $U_{t,l}$ – независимый и одинаково распределённый на полуинтервале [0,1) образец; τ – парметр температуры, если $\tau \to 0$, то имеем наиболее точное приближение ожидаемой случайной величины из распределения Гамбеля к ожидаемому значению категориальной случайной величины, а в случае $\tau \to \infty$ – ожидаемое значение из

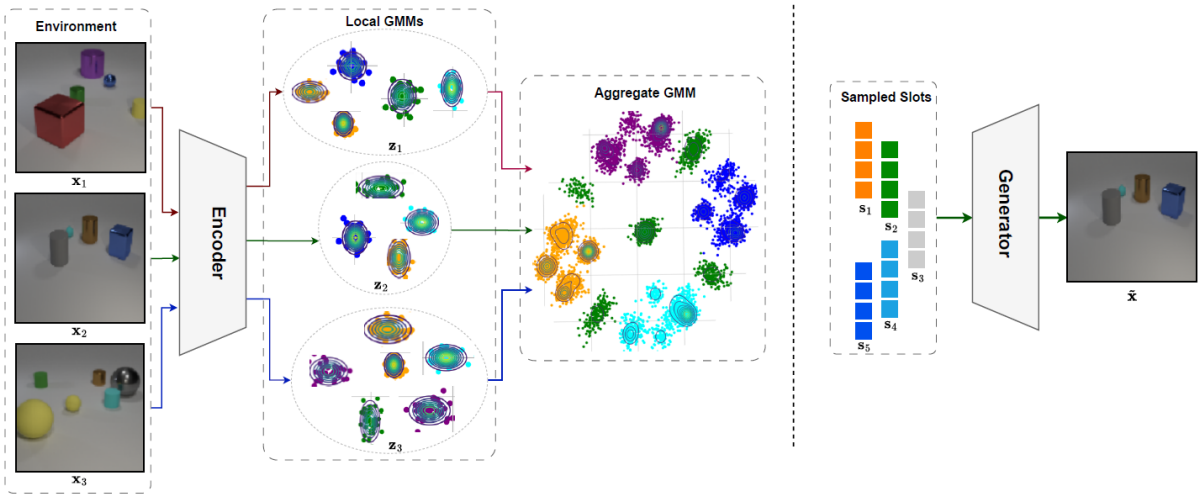


Рис. 3: Архитектура Probabilistic Slot Attention.

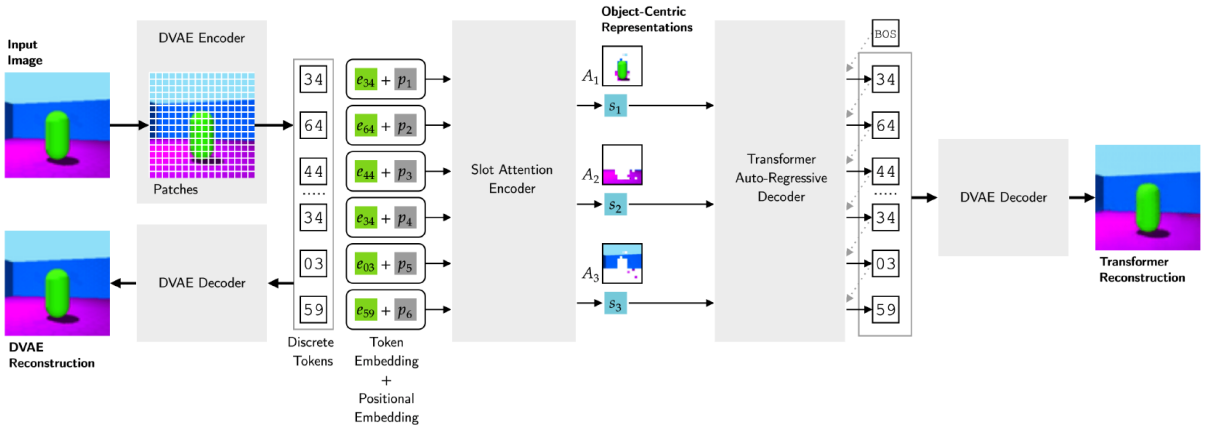


Рис. 4: Архитектура SLATE.

распределения сходится к образцам из равномерного распределения по категориям; $-\log(-\log U_{t,l})$ – шум из распределения Гамбеля. Функция softmax используется как непрерывное, дифференцируемое приближение $\arg\max$.

Для кодирования логарифмов вероятностей принадлежности к классу релаксированного категориального распределения $\mathbf{x}_{t,l}$ с понижением температуры τ от 1.0 до 0.1, а также реконструкции исходного изображения $\mathbf{x}_t$, строятся отображения $f_{\mathbf{enc}}$ и $f_{\mathbf{dec}}$:

$$\mathbf{o}_{t,l} = f_{\mathbf{enc}}(\mathbf{x}_{t,l}), \qquad \mathbf{z}_{t,l}^{\text{soft}} \sim \text{RelaxedCategorical}(\mathbf{o}_{t,l}; \tau), \qquad \hat{\mathbf{x}}_{t,l} = f_{\mathbf{dec}}(\mathbf{z}_{t,l}^{\text{soft}}), \tag{3.2}$$

где $\mathbf{o}_{t,l}$ – логарифм вероятности принадлежности к классу $\mathbf{v}$ образца из категориального распределения с V классами для l-го участка.

Первая часть задачи минимизации формулируется следующим образом:

$$\mathcal{L}_{\text{DVAE}} = \sum_{l=1}^{L} \|\mathbf{x}_{t,l} - \hat{\mathbf{x}}_{t,l}\|_2^2. \tag{3.3}$$

Библиотека визуальных концепций. Для создания библиотеки повторно используемых визуальных концепций, авторы используют подход, основанный на кластеризации $\mathcal{K}$-средних. Они собирают все K слотов $\mathcal{S}_t = \{\mathbf{s}_1^t, \ldots, \mathbf{s}_K^t\}$ и их карты внимания $\mathcal{A}_t = \{\mathcal{A}_1^t, \ldots, \mathcal{A}_K^t\}$ для каждого изображения. Далее запускается алгоритм $\mathcal{K}$-средних на $\mathcal{S} = \cup_t \mathcal{S}_t$, используя в качестве дистанционной метрики косинусное сходство, где $\mathcal{K}$ – количество всех концепций в библиотеке. Если положение объекта более важно для построения библиотеки, то слоты кластеризуются с применением IoU между картами внимания в качестве метрики сходства. Каждый кластер представляет собой концепцию, а слоты, назначенные кластеру – экземпляры концепций. Для создания произвольного изображения, выбирается концепция из библиотеки и случайным образом извлекается слот для каждой концепции. Таким образом формируется запрос, аналогичный текстовому в DALL-E, и он предоставляется декодеру для генерации изображения.

Slot Attention Encoder. Используя энкодер DVAE, для каждого участка l получают дискретный токен $\mathbf{z}_{t,l}$. Из обучаемого словаря визуальных концепций Dict_φ случайным образом на основе токенов $\mathbf{z}_{t,l}$ выбирается эмбеддинг l-го участка. Окончательный эмбеддинг $\mathbf{h}_{t,l} \in \mathcal{H}$, подаваемый на вход модулю Slot Attention (описан ранее), также учитывает информацию о позиции каждого токена $\mathbf{pos}_{\varphi,t,l}$:

$$\mathbf{z}_{t,l} \sim \text{Categorical}(\mathbf{o}_{t,l}),\ \mathbf{h}_{t,l} = \text{Dict}_\varphi(\mathbf{z}_{t,l}) + \mathbf{pos}_{\varphi,t,l},\ \mathcal{S}_t, \mathcal{A}_t = \text{SlotAttention}_\varphi(\mathcal{H}_t). \tag{3.4}$$

Transformer декодер. Полученные слотовые представления используются для реконструкции исходных изображений, для чего дополнительно строится ещё одно отображение $f_{\mathbf{Tr}}$:

$$\hat{\mathbf{o}}_{t,l} = f_{\mathbf{Tr}}(\hat{\mathbf{h}}_{t,<l}, \mathcal{S}_t), \quad \hat{\mathbf{z}}_{t,l} = \underset{\mathrm{v}\in[1,\mathrm{V}]}{\arg\max}\, \hat{\mathbf{o}}_{t,l,\mathrm{v}}, \quad \hat{\mathbf{x}}_{t,l} = f_{\mathbf{dec}}(\hat{\mathbf{z}}_{t,l}), \quad \mathcal{L}_{\mathrm{CE}} = -\sum_{l=1}^{L} \mathbf{z}_{t,l}\hat{\mathbf{o}}_{t,l}, \tag{3.5}$$

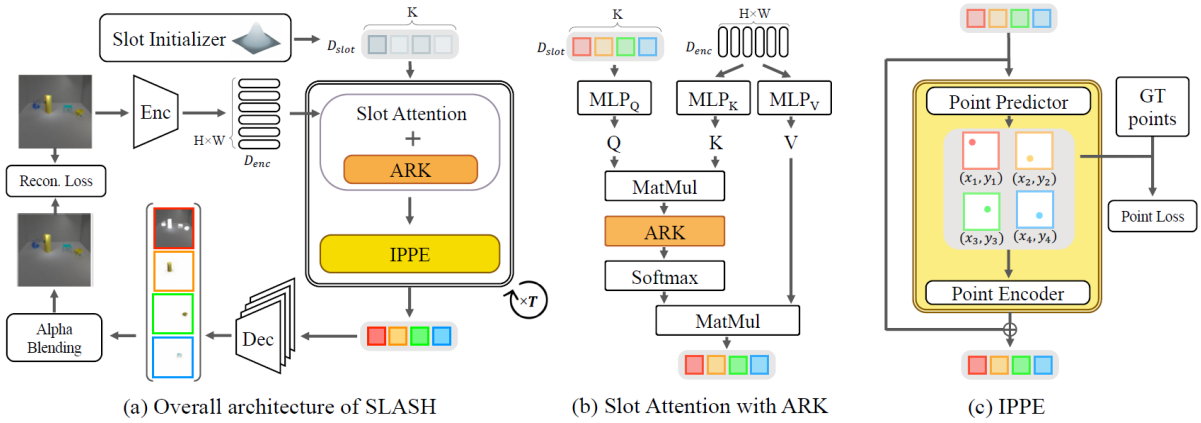
где $\hat{\mathbf{h}}_{t,<l} = \mathrm{Dict}_{\varphi}(\hat{\mathbf{z}}_{t,<l}) + \mathbf{pos}_{\varphi,t,<l}$; $\hat{\mathbf{o}}_{t,l,\mathrm{v}}$ – предсказание логарифма вероятности принадлежности токена к классу $\mathbf{v}$ среди $\mathbf{V}$ классов категориального распределения.

Окончательная задача – минимизировать следующий функционал:

$$\mathcal{L} = \min_{\hat{\mathbf{o}}_{t,l}\in\hat{\mathbb{O}},\ \hat{\mathbf{x}}_{t,l}\in\hat{\mathbb{X}}} \sum_{t=1}^{T} \mathcal{L}_{\mathrm{DVAE}}(\mathbf{x}_{t,l}, \hat{\mathbf{x}}_{t,l}) + \mathcal{L}_{\mathrm{CE}}(\mathbf{z}_{t,l}, \hat{\mathbf{o}}_{t,l}). \tag{3.6}$$

SLATE тестируется на наборах данных: 3D Shapes [15], CLEVR-Mirror [11], Shapestacks [18], Textured MNIST, CLEVRTex [31] и CelebA – все они представляют собой наборы изображений. Модель оценивается по MSE и FID (Fréchet Inception Distance), в целом демонстрируя хорошие результаты в генерации и реконструкции изображений, в превалирующем большинстве случаев опережая базовое решение – Slot Attention. Главным и очень сильным ограничением для модели является то, что она не использует временную информацию, т.е. отсутствует поддержка согласованной идентичности объектов во времени, невозможно её отслеживание, а, соответственно, модель не может масштабироваться на видеоданные. Хотя на уровне кадра демонстрируется хорошая сегментация.

SLASH. В 2023-м году был представлен SLASH [40], где авторы сосредотачиваются на проблеме объектно-ориентированных представлений для single-view изображений, где предоставляется меньше информации об объекте и его фоне, чем в multi-view изображения и видео, т.е. когда мы можем посмотреть на изучаемые объекты с разных точек зрения, под разными углами и на различном расстоянии. Single-view изображения имеют проблему, заключающуюся в том, что для них сложно получить индуктивные искажения, основанные на данных, таких как геометрическая или временная информация. Однако в последнее время были разработаны архитектуры, представленные ранее, которые решают эту проблему, но сложные изображения требуют дополнительных индуктивных искажений и продолжают страдать от проблем, связанных с нестабильностью результатов обучения.



(a) Overall architecture of SLASH (b) Slot Attention with ARK (c) IPPE

Рис. 5: Архитектура SLASH.

Проблема заключается в утечке внимания, которое приводит к тому, что в итоге получаются объектно-ориентированные представления, основанные на искаженных картах внимания. Данная проблема является очень значимой, поскольку трудно предсказать поведение модели, т.е. захватит ли слот отдельный объект или же объект, запутанный в фоне (с некоторым шумом). SLASH, как утверждают авторы статьи, решает данную проблему, направляя случайно инициализированные слоты для успешного захвата объектов: 1) не отвлекаясь на фон и 2) сохраняя информацию о таргете (цели). Это достигается добавлением двух простых, но эффективных модулей Attention Refining Kernel (ARK) и Intermediate Point Predictor and Encoder (IPPE) для модуля Slot Attention. Более подробно архитектура модели изображена на рисунке 5.

Attention Refining Kernel. ARK – это одноканальное однослойное сверточное ядро, предназначенное для предотвращения фокусировки слотов на зашумленном фоне. Используется сверточное ядро с нормализованными весами WNConv – обучаемый фильтр нижних частот в качестве ядра для ARK. Это простое ядро улучшает распределение внимания между слотами и пикселями за счет уменьшения шума и закрепления объектоподобных паттернов.

Intermediate Point Predictor and Encoder. IPPE служит индикатором для перемещения слота, чтобы сфокусироваться на нужном местоположении. Таким образом, слоты могут последовательно обновлять свои представления, не обращая внимания на фон. IPPE состоит из двух подмодулей с простыми MLP. Первый подмодуль предсказывает положение объекта в двумерных координатах, а второй – кодирует предсказанные координаты в многомерный вектор.

Чтобы обучить IPPE предоставлять сигналы местоположения для слотов, необходимо ввести позиционные метки. Однако использование полностью аннотированных исходных данных обходится дорого, особенно для меток с плотными аннотациями, таких как маски объектов. Поэтому применяется weakly semi-supervised подход, при котором только небольшое подмножество набора данных включает слабые аннотации, такие как центры ограничивающих рамок. Авторы показывают, что этот модуль может быть успешно обучен и развёрнут в условиях, когда не существует вспомогательных истинных подсказок.

Более формально, задан набор изображений $\mathbf{x}_t \in \{\mathbf{x}_1, ..., \mathbf{x}_T\}$, ставится задача построения фильтра нижних частот $f_{\mathbf{ARK}}$ для предотвращения утечки внимания:

$$\mathbf{x}_t^* = f_{\mathbf{enc}}(\mathbf{x}_t), \quad \mathbf{M}_t = \frac{1}{\sqrt{D}} k(\mathbf{x}_t^*) \cdot q(\mathcal{S}_t)^T, \quad \mathcal{A}_t = \underset{K}{\text{softmax}}\left(f_{\mathbf{ARK}}(\mathbf{M}_t)\right) \in \mathbb{R}^{N \times K}, \quad (3.7)$$

$$\mathcal{U}_t = \frac{1}{\mathcal{W}_t} \sum_{n=1}^{N} \mathcal{A}_{t,n} \odot v(\mathbf{x}_{t,n}^*) \in \mathbb{R}^{K \times D}, \quad \hat{s}_t^k = \text{GRU}(u_t^k, s_t^k) \in \hat{\mathcal{S}}_t. \quad (3.8)$$

Дополнительно строится отображение $f_{\mathbf{IPPE}}$ для информирования слотов о положении важных областей $\mathbf{c}_r$ в виде пары координат на изображении:

$$\hat{s}_t^k, \mathbf{c}_r, \hat{\mathbf{c}}_r = f_{\mathbf{IPPE}}(\hat{s}_t^k), \quad \hat{s}_t^k \leftarrow \hat{s}_t^k + \text{MLP}(\text{LN}(\hat{s}_t^k)). \quad (3.9)$$

Обновлённые слотовые представления используются для реконструкции $\hat{\mathbf{x}}_t$ исходного изображения и его альфа-маски $\hat{\mathbf{m}}_t$:

$$\hat{\mathbf{m}}_t^k, \hat{\mathbf{x}}_t^k = f_{\mathbf{dec}}(\hat{s}_t^k), \quad \mathbf{m}_t = \underset{K}{\text{softmax}}(\hat{\mathbf{m}}_t^k), \quad \hat{\mathbf{x}}_t = \sum_{k=1}^{K} \mathbf{m}_t^k \odot \hat{\mathbf{x}}_t^k. \quad (3.10)$$

Минимизируются функционалы ошибки по точкам и восстановления:

$$\mathcal{L}_{points} = \min_{\hat{\mathbf{c}}_k \in \mathcal{C}} \sum_{k=1}^{K} \|\mathbf{c}_k - \hat{\mathbf{c}}_k\|_2^2, \quad (3.11)$$

$$\mathcal{L}_{\text{rec}} = \min_{\hat{\mathbf{x}}_t \in \hat{\mathbb{X}}} \sum_{t=1}^{T} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2, \quad (3.12)$$

где $\hat{\mathbf{c}}_r$ – предсказание координат области r; $\mathcal{R}$ – множество важных областей.

Используются три метрики: mIoU, ARI и FG-ARI. Первые две исследуют, возникает ли проблема утечки внимания, рассматривая разделение фона. Модель тестируется на наборах данных: CLEVR6 [11], CLEVRTEX [31], PTR [32] и MOVi-C [35]. В качестве базовых решений были выбраны: Slot Attention [25], GENESIS-v2 [30] и

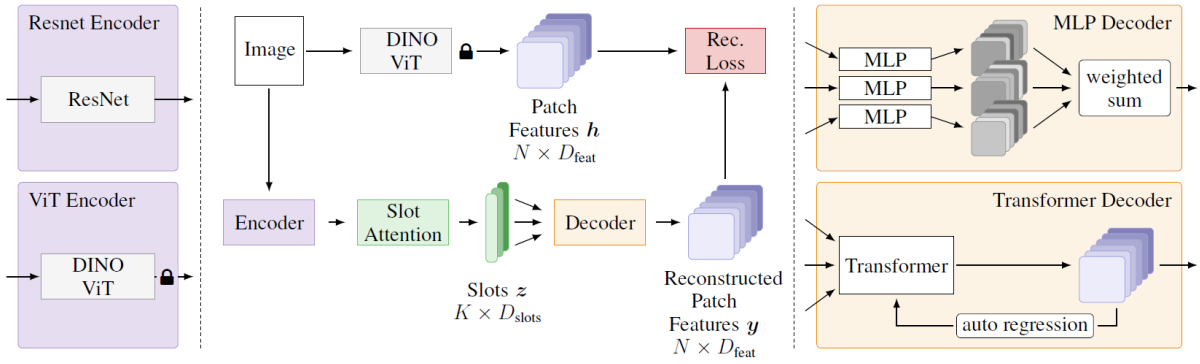


Рис. 6: Архитектура DINOSAUR.

weakly supervised Slot Attention. Авторы демонстрируют, что SLASH показывает наилучшую стабильность, превосходя предыдущие методы для всех наборов данных по нескольким показателям.

DINOSAUR. В 2023-м году была предложена интересная с точки зрения идеи модель на основе self-supervised подхода к обучению – DINOSAUR [39]. Она обучается восстанавливать не исходное изобраажение, а некоторые закодированные признаки этого изображения с помощью предобученного DINO ViT [29], выходы которого являются предсказываемыми значениями. С архитектурой DINOSAUR можно ознакомится на рисунке 6.

Авторы предполагают, что реконструкция на уровне пикселей даёт слишком слабый сигнал для проявления объектной ориентированности, т.к. задача фокусируется (по крайней мере, на начальном этапе) в основном на низкоуровневых характеристиках изображения, таких как цветовая статистика. Этим они объясняют неудачи предыдущих моделей за пределами простых синтетических наборов данных. Вместо этого, если бы был сигнал, для восстановления которого требовалась бы семанитческая информация более высокого уровня, на слоты также оказывалось бы давления для эффективного кодирования этой информации. Размышляя в этом направлении, авторы решили воспользоваться возможностями self-supervised алгоритмов обучения, в частности DINO. Для предобучения модели учителя используется набор данных ImageNet [1]. С точки зрения ученика и учителя [6], это означает, что учитель дополнительно переносит знания, полученные из более крупной коллекции изображений, в имеющиеся (меньшие) наборы данных. Минимизируя среднеквадратическую ошибку, мы можем утверждать, что оптимизация происходит в пространстве более семантическом, чем пиксельное.

Энкодер. Предыдущие работы показывают, что мощные средства извлечения объектов помогают масштабировать объектно-ориентированные методы на более сложные данные. Поэтому авторы экспериментировали с энкодерами ResNet-34 [8] и предобученным ViT [27]. Они пришли к выводу, что оба энкодера работают аналогично, однако модель сходится быстрее с ViT, а также более эффективна в вычислительном отношении (можно использовать выходные данные ViT в качестве целевых сигналов). Поэтому на его основе строится отображение $f_{\mathbf{enc}}$:

$$\mathbf{x}_t^* = f_{\mathbf{enc}}(\mathbf{x}_t). \tag{3.13}$$

Slot Attention. В качестве модуля Slot Attention используется оригинальная реализация с одним отличием – авторы не добавляют позиционной кодировки к признакам ViT перед подачей в Slot Attention, поскольку обнаружили, что исходноых кодировок ViT достаточно для поддержки пространственной группировки объектов. Помимо этого, добавляется небольшой MLP с одним скрытым слоем, которые преобразует каждый закодированный признак перед Slot Attention. *Декодер.* В поисках архитектуры декодера для реконструкции признаков, авторы экспериментировали с MLP и Transformer декодерами. Результаты в целом получаются неоднозначные, но об этом будет написано далее. Итак, после получения слотовых представлений строится отображение $f_{\mathbf{dec}}$:

$$\hat{s}_t^k = \text{SlotAttention}(\mathbf{x}_t^*),\ \hat{\mathbf{m}}_t^k, \hat{\mathbf{f}}_t^k = f_{\mathbf{dec}}(\hat{s}_t^k),\ \mathbf{m}_t = \underset{K}{\text{softmax}}(\hat{\mathbf{m}}_t^k),\ \hat{\mathbf{f}}_t = \sum_{k=1}^{K} \mathbf{m}_t^k \odot \hat{\mathbf{f}}_t^k. \tag{3.14}$$

Минимизируется следующий функционал:

$$\mathcal{L}_{\text{rec}} = \min_{\hat{\mathbf{f}}_t \in \mathcal{F}} \sum_{t=1}^{T} \|\mathbf{f}_t - \hat{\mathbf{f}}_t\|_2^2, \tag{3.15}$$

где $\mathbf{f}_t$ – признаки, полученные от модели учителя (DINO ViT).

DINOSAUR тестируется на наборах: MOVi-C, MOVi-E [35], PASCAL VOC 2012 [2], COCO [5], KITTI [14]. В ходе проведённых экспериментов авторы показывают:

1. Реконструкция изображений не даёт достаточного сигнала для семантической группировки, даже если отталкиваться от признаков, которые обладают высокой семантикой.

2. Восстановление признаков является ключевым, поскольку позволяет проявлять объектную ориентированность на реальных данных.

3. Предобучение на наборе ImageNet приводит к полезному обобщению при решении задачи обнаружения объектов.

4. MLP декодер показывает лучшие метрики FG-ARI (Foreground Adjusted Rand Index), а Transformer декодер лучше по метрике mBO (mean Best Overlap). mBO вычисляется путём присвоения каждой истинной маске некоторой прогнозируемой маски с наибольшим перекрытием, а затем усреднения значений IoU's назначенных пар масок. FG-ARI – показатель, измеряющий сходство кластеров. MLP лучше разделяет объекты, что выражается в повышенном ARI. Transformer стремится создавать более плотные маски и более чёткое разделение фона, однако для разделения объектов используются лишние прорези.

4 Метод

В данной работе предлагается объектно-ориентированная модель RAPID (**R**obust **A**daptive **P**robabilistic **I**nference with **DI**NO features). Используется традиционный дизайн объектно-ориентированных моделей, похожий на автокодировщик. RAPID включает в себя 4 блока: энкодер на основе DINO Vision Transformer (ViT), параметризованный инициализатор слотовых представлений, стабилизированный Slot Mixture модуль выделения слотовых представлений и декодер на основе многослойного перцептрона (MLP). Со схемой можно ознакомиться на рисунке 7.

Подход основан на стабилизированной версии SMM. Ключевым отличием от классической формулировки SMM является дополнительное обновление оценки на апостериорное распределение коэффициентов псевдовнимания после каждой корректировки ключевых параметров смеси. Помимо этого, по сравнению с DINOSAUR, в нашем методе реализованы две основные архитектурные модификации: (1) иерархическая конкатенация (ИК) ключевых функций, извлеченных из кодировщика, и (2) модуль инициализатор на основе трансформера.

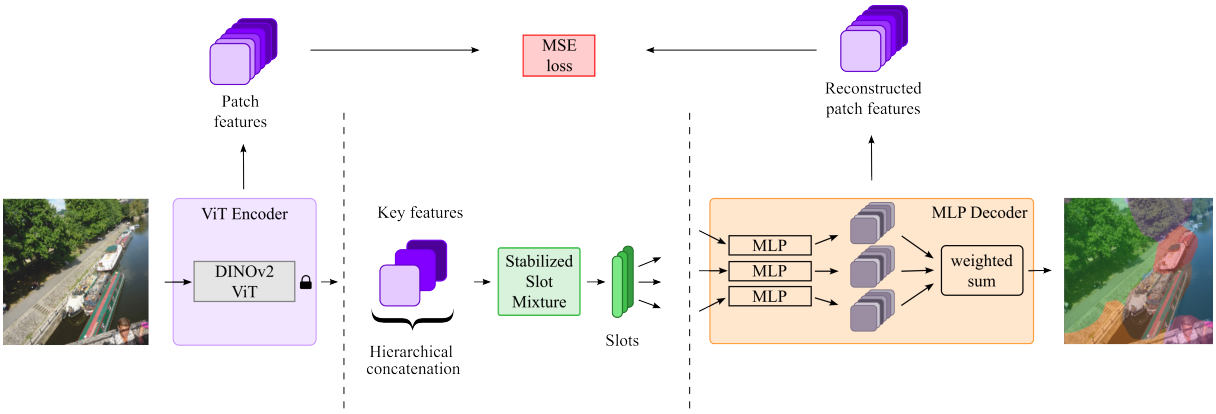


Рис. 7: Архитектура RAPID. Изображение разбивается на набор патчей $\mathbf{f}_t$ и обрабатывается с помощью ViT, предобученного с применением self-supervised алгоритма DINO на наборе ImageNet. Стабилизированный Slot Mixture модуль группирует объекты в набор слотов, а модель обучается путём восстановления DINO признаков из слотовых представлений, независимо для каждого слота.

4.1 Энкодер

Предыдущие работы демонстрируют, что мощные инструменты извлечения признаков помогают масштабировать объектно-ориентированные методы на более сложные данные. Поэтому мы используем ViT, предварительно обученный с помощью

алгоритма DINO [27] на ImageNet [1]:

$$\mathbf{f}_t = f_{\mathbf{enc}}(\mathbf{x}_t) \in \mathbb{R}^{N \times D_\text{f}}, \qquad \mathbf{f}_t^* = \text{Conv}(\text{Concat}[\mathbf{f}_t^{l-1}, \mathbf{f}_t^{l-2}, \mathbf{f}_t^{l-3}]) \in \mathbb{R}^{N \times D_\text{s}}, \tag{4.1}$$

where f_enc – замороженный энкодер ViT, предварительно обученный по алгоритму DINO; l – количество блоков в ViT; N – количесвто токенов; D_f, D_s – размерность признаков и слотов, соответственно; $\mathbf{f}_t$ – признаки, полученные с энкодера DINO ViT; $\mathbf{f}_t^*$ – признаки, полученные после ИК.

4.2 Инициализатор слотовых представлений

Данный модуль принимает в качестве входных данных признаки, полученные с энкодера, и инициализирует представления слотов. В качестве начального приближения для параметров алгоритма μ и Σ используется нулевое среднее значение, а инициализация параметра логарифмического стандартного отклонения производится из равномерного распределения по методу Ксавье. Впоследствии эти параметры уточняются с помощью обучаемых блоков трансформера и MLP. Блоки трансформера используют механизм self-attention, а MLP состоит из пары линейных слоев с нелинейной функцией активации ReLU между ними. Гипотеза использования такого инициализатора состоит в том, что вместо добавления случайного шума к фиксированным значениям μ и Σ мы можем использовать контекстную информацию из входных данных для генерации более обоснованных начальных значений для слотов.

4.3 Модуль выделения слотовых представлений

Slot Mixture Module. Мы стремимся построить вероятностную модель, основанную на предположении о правдоподобии, учитывая, что все объекты выборки распределены независимо и одинаково. Следовательно, вероятность $P(\mathbb{X} \mid \theta)$ наблюдаемых данных $\mathbb{X} = \{x_1, \ldots, x_T\}$ с заданными параметрами θ в непрерывном случае может быть выражена как произведение функций плотности вероятности $f(x_t \mid \theta)$, а в дискретном случае — произведением вероятностей $p(\mathbf{x}_t \mid \theta)$. Более формально, мы хотим построить вероятностную модель для экземпляра $\mathbf{x}_t \in \mathbb{X}$, порожденного смесью распределений с K компонентами и их априорными весами $\pi \in \mathbb{R}^K$:

$$\mathbf{x}_t \sim p(\mathbf{x}_t \mid \theta) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}_t \mid \theta_k), \quad P(\mathbb{X} \mid \theta) = \prod_{t=1}^{T} p(\mathbf{x}_t \mid \theta), \quad \sum_{k=1}^{K} \pi_k = 1, \tag{4.2}$$

где θ_k – параметры k-й компоненты смеси.

Мы предполагаем, что каждый компонент смеси моделируется гауссовым распределением со своими параметрами, а именно средним значением μ и ковариационной матрицей Σ, которую мы считаем диагональной. Используя EM алгоритм, ищется K групп параметров θ_k, которые определяют каждый экземпляр $\mathbf{x}_t$ и максимизируют оценку правдоподобия модели $P(\mathbb{X} \mid \theta)$.

E step:

$$P(\mathbb{X}, \mathbb{X}^* \mid \theta^*) = \prod_{t=1}^{T} \prod_{k=1}^{K} [\pi_k p(\mathbf{x}_t \mid \theta_k^*)]^{I(\mathbf{x}_t^* = \mathbf{x}_k^*)}, \tag{4.3}$$

$$Q(\theta^*, \pi^* \mid \theta, \pi) = \mathbb{E}_{P(\mathbb{X}^* \mid \mathbb{X}, \theta)}[\log P(\mathbb{X}, \mathbb{X}^* \mid \theta^*)], \tag{4.4}$$

$$p(\mathbf{x}_k^* \mid \mathbf{x}_t) = \frac{p(\mathbf{x}_k^*) p(\mathbf{x}_t \mid \theta_k)}{\sum_{k=1}^{K} p(\mathbf{x}_k^*) p(\mathbf{x}_t \mid \theta_k)} = \frac{\pi_k \mathcal{N}(\mathbf{x}_t \mid \mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_t \mid \mu_k, \Sigma_k)} = \gamma_{k,t}, \tag{4.5}$$

$$\mathcal{A}_{k,t} = \frac{\gamma_{k,t}}{\sum_{d=1}^{D} \gamma_{d,k,t}}, \tag{4.6}$$

где $\mathbb{I}[\cdot]$ – индикаторная функция; $\gamma_{k,t}$ and $\mathcal{A}_{k,t}$ – ненормализованные и нормализованные, соответственно, коэффициенты псевдовнимания для k-го слота t-го экземпляра.

M step:

$$(\theta, \pi) = \arg\max_{(\theta^*, \pi^*)} Q(\theta^*, \pi^* \mid \theta, \pi), \tag{4.7}$$

$$\pi_k^* = \frac{\sum_{t=1}^{T} \mathcal{A}_{k,t}}{T}, \qquad \mu_k^* = \frac{\sum_{t=1}^{T} \mathcal{A}_{k,t} \mathbf{x}_t}{\sum_{t=1}^{T} \mathcal{A}_{k,t}}, \tag{4.8}$$

$$\Sigma_k^* = \frac{\sum_{t=1}^{T} \mathcal{A}_{k,t} (\mathbf{x}_t - \mu_k)(\mathbf{x}_t - \mu_k)^T}{\sum_{t=1}^{T} \mathcal{A}_{k,t}}. \tag{4.9}$$

SMM обновляет свои параметры, максимизируя оценку функции правдоподобия модели (4.4).

Stabilized Slot Mixture Module. Вместо классического Slot Attention и Slot Mixture мы представляем новую стабилизированную версию Slot Mixture модуля (SSMM), схематическое изображение которого показано на рисунке 8. Ключевая идея предлагаемого улучшения заключается в том, что после обновления ключевых параметров смеси также обновляется оценка на апостериорное распределение коэффициентов псевдовнимания γ, которые играют решающую роль в модели. В алгоритме 4.1 logits и γ представлены в $\mathbb{R}^{K \times D_{\text{f}}}$. В 11-й строке мы применяем операцию LayerNorm к параметру μ, чтобы обеспечить согласованность масштаба. После обновления каждого из параметров π, μ и Σ_{diag} мы приступаем к корректировке оценки

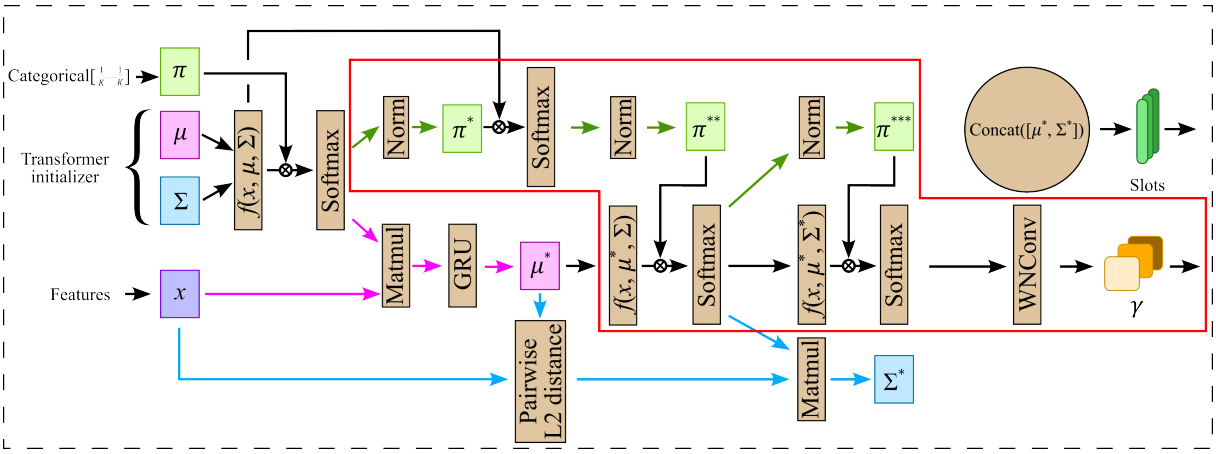


Рис. 8: Стабилизированный Slot Mixture модуль. Добавленные архитектурные блоки находятся внутри красного контура.

на апостериорное распределение (как подробно описано в строках 6-8, 12-15, 18-20 алгоритма 4.1). После обновления диагональной ковариационной матрицы Σ_{diag} распределение коэффициентовγ корректируется исключительно на последней итерации алгоритма. Кроме того, мы используем свертку с нормализованными весами ядра [40] на картах ненормализованных коэффициентов псевдовнимания γ для достижения более сглаженных и менее «шумных» представлений.

Мы предполагаем, что значительное и резкое снижение качества представлений классического SMM связано с отсутствием необходимых корректировок при обновлении параметров. В частности, обновления ключевых параметров смеси основаны на оценках, полученных на E шаге, которые могут утратить актуальность после обновления параметров. Следовательно, предлагается проводить повторную оценку на апостериорное распределение коэффициентов псевдовнимания γ после обновления каждого из ключевых параметров смеси (π, μ, Σ) с использованием обновлённых параметров смеси. Эта модификация классического алгоритма не только устраняет проблему резкого снижения качества, но и повышает качество модели.

4.4 Декодер

Мы используем MLP-декодер, который повторяет конструкцию spatial broadcast декодера [21]. Каждый слот сначала транслируется на определенное количество патчей, восстанавливающих набор признаков из N токенов. Чтобы сделать пространственное расположение токенов идентифицируемым, к каждому токену добавляется обучаемая позиционная кодировка. Токены для каждого слота затем обрабатывают-

Алгоритм 4.1 Стабилизированный Slot Mixture модуль.

Вход: $\mathbf{f}^* \in \mathbb{R}^{N \times D_\text{s}}$ – DINO признаки после ИК; μ, $\Sigma \in \mathbb{R}^{K \times D_s}$, $\pi \in \mathbb{R}^K$ – параметры и веса смеси; $f_\theta(\cdot)$ – функция ядра нормального распределения; $\mathcal{I} = \{1, \ldots, L\}$.

Выход: slots $\in \mathbb{R}^{N \times 2D_\text{s}}$, $\gamma_\text{new} \in \mathbb{R}^{K \times D_\text{f}}$.

для $i \in \mathcal{I}$
  $\text{logits} = f_\theta(\mathbf{f}^*, \mu, \Sigma)$,
  $\gamma = \text{Softmax}(\text{logits} + \log \pi, \text{dim} = 0)$,
  $\mathcal{A} = \text{Normalize}(\gamma, \text{dim} = 1)$,
  $\pi^* = \mathcal{A}.\text{mean}(\text{dim} = 0)$,
  $\boxed{\gamma^* = \text{Softmax}(\text{logits} + \log \pi^*, \text{dim} = 0)}$,
  $\boxed{\mathcal{A}^* = \text{Normalize}(\gamma^*, \text{dim} = 1)}$,
  $\boxed{\pi^{**} = \mathcal{A}^*.\text{mean}(\text{dim} = 0)}$,
  $\mu^* = \text{WeightedMean}(\text{weights} = \mathcal{A}^*, \text{values} = \mathbf{f}^*)$,
  $\mu^* = \text{GRU}(\text{input} = \mu^*,\ \text{hidden} = \mu)$,
  $\mu^* = \text{MLP}[\text{LayerNorm}(\mu^*)] + \boxed{\text{LayerNorm}(\mu^*)}$,
  $\boxed{\text{logits}^* = f_\theta(\mathbf{f}^*, \mu^*, \Sigma)}$,
  $\boxed{\gamma^{**} = \text{Softmax}(\text{logits}^* + \log \pi^{**}, \text{dim} = 0)}$,
  $\boxed{\mathcal{A}^{**} = \text{Normalize}(\gamma^{**}, \text{dim} = 1)}$,
  $\boxed{\pi^{***} = \mathcal{A}^{**}.\text{mean}(\text{dim} = 0)}$,
  $\Sigma^*_\text{diag} = \text{WeightedMean}(\text{weights} = \gamma^{**}, \text{values} = (\mathbf{f}^* - \mu^*)^2)$,
  если $i = L$ **то**
    $\boxed{\text{logits}^{**} = f_\theta(\mathbf{f}^*, \mu^*, \Sigma^*)}$,
    $\boxed{\gamma^{***} = \text{Softmax}(\text{logits}^{**} + \log \pi^{***}, \text{dim} = 0)}$,
    $\boxed{\gamma_\text{new} = \text{WNConv}(\gamma^{***})}$,
$\text{slots} = \text{concat}([\mu^*, \Sigma^*])$,
return slots, γ_new,

ся по токенам одним и тем же MLP, а результатом является реконструкция $\hat{\mathbf{f}}_{k,t}$ для слота k и альфа-маски $\hat{\mathbf{m}}_{k,t}$. Конечный результат реконструкции $\hat{\mathbf{f}}_t \in \mathbb{R}^{N \times D_\text{f}}$ представляет собой взвешенную сумму по слотам:

$$\hat{\mathbf{m}}_{k,t}, \hat{\mathbf{f}}_{k,t} = f_{\mathbf{dec}}(s_{k,t}), \quad \mathbf{m}_t = \underset{K}{\text{Softmax}}(\hat{\mathbf{m}}_{k,t}), \quad \hat{\mathbf{f}}_t = \sum_{k=1}^{K} \mathbf{m}_{k,t} \odot \hat{\mathbf{f}}_{k,t}. \tag{4.10}$$

Минимизируется следующий функционал:

$$\mathcal{L}_\text{rec} = \sum_{t=1}^{T} \|\mathbf{f}_t - \hat{\mathbf{f}}_t(\theta)\|_2^2. \tag{4.11}$$

5 Эксперименты

5.1 Наборы данных

Мы рассматриваем три набора данных: PASCAL VOC 2012 [2], MS COCO 2017 [5] и ClevrTex [31]. Обзор наборов данных представлен в таблице 1. В то время как PASCAL VOC содержит множество изображений только с одним крупным объектом, COCO состоит из изображений как минимум с двумя, а часто и с десятками объектов. Оба набора данных представляют собой значительный шаг вперед по сложности по сравнению с тем, на чем до сих пор тестировались объектно-ориентированные модели. ClevrTex предлагает синтетические сцены с разнообразными формами, текстурами и фотоматериалами, созданными с использованием физических методов рендеринга. На изображениях данного набора обычно изображено 3-10 объектов.

Таблица 1: Обзор наборов данных.

	COCO	PASCAL VOC	ClevrTex
Train Size	118287	10582	35000
Validation Size	5000	2913	5000
Image/Crop Size	224	224	224
Random Horizontal Flip	✓	✓	✗
Train Crop Type	Random	Random	Full
Validation Crop Type	Center	Center	Full

5.2 Детали обучения

RAPID обучается с использованием оптимизатора Adam [3] и пиковым коэффициентом скорости обучения (LR) 4×10^{-4}. Расписание обучения состоит из трех этапов: начальная линейный разгон в течение 5×10^3 итераций, за которым следует косинусоидальное затухание, снижающее значение LR до $0,2$ от его первоначального значения в течение 5×10^4 итераций. На заключительном этапе значение коэффициента скорости обучения снижается по экспоненицальному закону. Полный список гиперпараметров приведен в таблице 2.

Таблица 2: Гиперпараметры для RAPID, используемые на наборах COCO, PASCAL и ClevrTex.

		COCO	**PASCAL**	**ClevrTex**
Количество итераций обучения		200k	200k	100k
Размер батча		64	64	64
Обрезка нормы градиента		1.0	1.0	1.0
Пиковый LR		0.0004	0.0004	0.0004
Шаги разгона LR		5000	5000	5000
Минимальный коэф. масштабирования LR		0.2	0.2	0.2
Коэф. экспоненциального затухания		0.65	0.65	0.65
Архитектура ViT		ViT-B	ViT-B	ViT-B
Размер патча		14	14	8
Разм. признаков, D_{f}		768	768	768
SSMM	Slots	7	6	11
	Кол-во итераций	3	3	3
	Разм. слотов, $D_{\mathrm{s}} = D_\mu + D_\Sigma$	512	512	512
	Скрытая разм. MLP	2048	2048	2048
	Выходная разм.	256	256	256
Декодер	Тип	MLP	MLP	MLP
	Кол-во слоёв	4	4	4
	Скрытая разм.	2048	2048	2048

5.3 Результаты

Мы оцениваем качество моделей, используя Foreground Adjusted Rand Index (FG-ARI) [20], показатель, который измеряет сходство кластеров, игнорируя фон, и классический ARI:

$$\mathrm{RI} = \frac{2(a+b)}{n(n-1)}, \qquad \mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max \mathrm{RI} - \mathbb{E}[\mathrm{RI}]}, \tag{5.1}$$

где a — число пар объектов, имеющих одинаковые метки в обоих разбиениях (кластеризациях) $\mathbb{Y}$ и $\hat{\mathbb{X}}$; b — число пар объектов, имеющих различные метки в обоих разбиениях; n — число объектов в выборке.

Таблица 3: Сравнение с DINOSAUR и DINO SMM (среднее ± стандартное отклонение, 3 сида) с 6 (PASCAL), 7 (COCO) и 11 (ClevrTex) слотами.

		FG-ARI	mBO
COCO	DINOSAUR	40.5 ± 0.2	27.7 ± 0.2
	DINO SMM	50.4 ± 0.2	28.1 ± 0.1
	RAPID (Мы)	$\mathbf{52.3 \pm 0.2}$	$\mathbf{30.5 \pm 0.1}$
PASCAL	DINOSAUR	24.6 ± 0.2	40.9 ± 0.1
	DINO SMM	$\mathbf{27.1 \pm 0.1}$	$\mathbf{43.5 \pm 0.3}$
	RAPID (Мы)	$\mathbf{27.0 \pm 0.1}$	$\mathbf{43.7 \pm 0.1}$
ClevrTex	DINOSAUR	64.8 ± 0.1	30.9 ± 0.1
	DINO SMM	84.2 ± 0.2	51.3 ± 0.3
	RAPID (Мы)	$\mathbf{96.7 \pm 0.1}$	$\mathbf{62.0 \pm 0.3}$

Кроме того, мы вычисляем показатель, основанный на пересечении через объединение (IoU), называемый mean Best Overlap (mBO) [13]. mBO оценивает, насколько хорошо маски соответствуют объектам, и рассчитывается путем присвоения каждой основной маске соответствующей прогнозируемой маски с наибольшим перекрытием, а затем усреднения значений назначенных пар масок:

$$\text{mBO} = \frac{1}{TC} \sum_{t=1}^{T} \sum_{c=1}^{C} \frac{\mathbf{y}_{c,t} \cap \arg\max_{\hat{\mathbf{x}}_{c,t} \in \hat{\mathbb{X}}} \mathbf{y}_{c,t} \cap \hat{\mathbf{x}}_{c,t}}{\mathbf{y}_{c,t} \cup \arg\max_{\hat{\mathbf{x}}_{c,t} \in \hat{\mathbb{X}}} \mathbf{y}_{c,t} \cap \hat{\mathbf{x}}_{c,t}}, \tag{5.2}$$

где C — количество классов; $\mathbf{y}_{c,t}$ — истинные метки для экземпляра с порядковым номером t в классе c; $\hat{\mathbf{x}}_{c,t}$ — оценки для экземпляра t в классе c.

Наш метод демонстрирует значительно лучшие результаты, чем выбранные бейзлайны, в двух из трех наборов данных — COCO и ClevrTex — по обоим показателям, результаты представлены в таблице 3. Визуализацию масок можно видеть на рисунках 9, 10 и 11. Несмотря на то, что наш метод демонстрирует разницу по сравнению с базовыми подходами в нескольких тестах как с точки зрения FG-ARI, так и с точки зрения mBO, мы наблюдаем небольшое отставание в показателях FG-ARI в наборе данных PASCAL VOC от классического SMM. Это несоответствие особенно интригует, учитывая, что на данном наборе наш метод превосходит бейзлайны по метрике mBO и хорошо работает с другими наборами данных. Предполагается, что такое поведение тесно связано с характеристиками самого набора данных PASCAL

VOC. В отличие от более современных или синтетических наборов данных с более четкими аннотациями и более однородным распределением объектов, PASCAL VOC содержит изображения реального мира со значительными различиями в масштабе объектов, окклюзии и сложности фона. Подводя итог, мы считаем, что наблюдаемая динамика производительности в PASCAL VOC отражает ограничения, характерные для конкретного набора данных, а не недостаток нашего метода. Эта интерпретация подтверждается нашими неизменно высокими результатами на других, более тщательно отобранных или синтетически сгенерированных наборах данных, где аннотации более точно соответствуют предположениям о задачах сегментации объектов без учителя.



Рис. 9: Визуализация примеров из набора COCO.



Рис. 10: Визуализация примеров из набора COCO.



Рис. 11: Визуализация примеров из набора COCO.

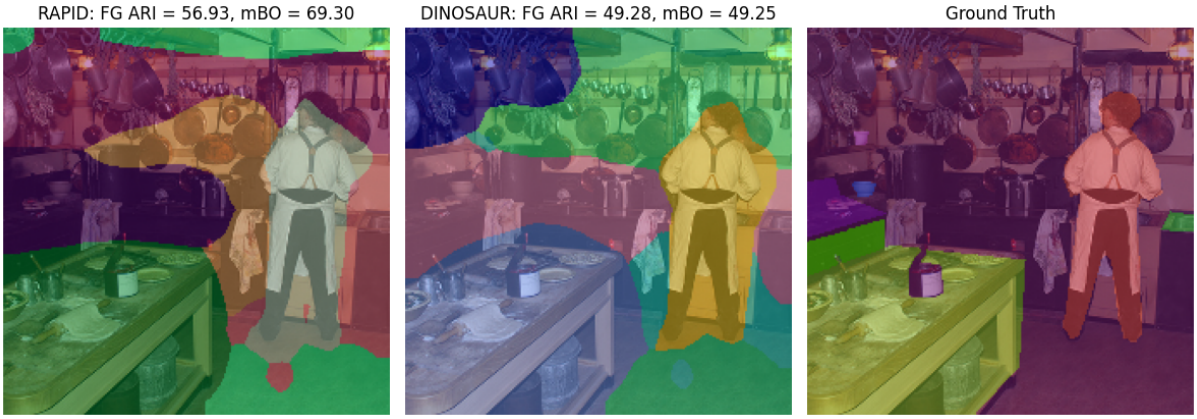


Рис. 12: Визуализация примеров из набора COCO.



Рис. 13: Визуализация примеров из набора COCO.

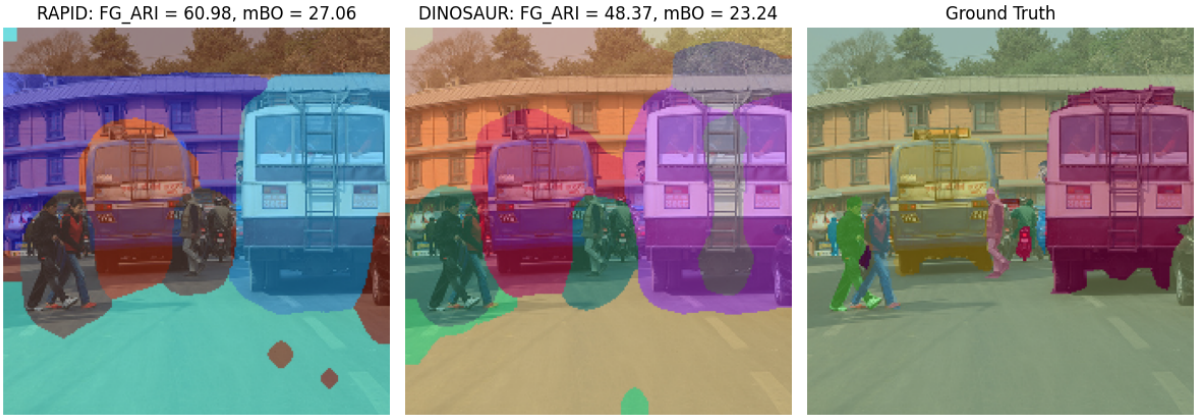


Рис. 14: Визуализация примеров из набора COCO.



Рис. 15: Визуализация примеров из набора COCO.

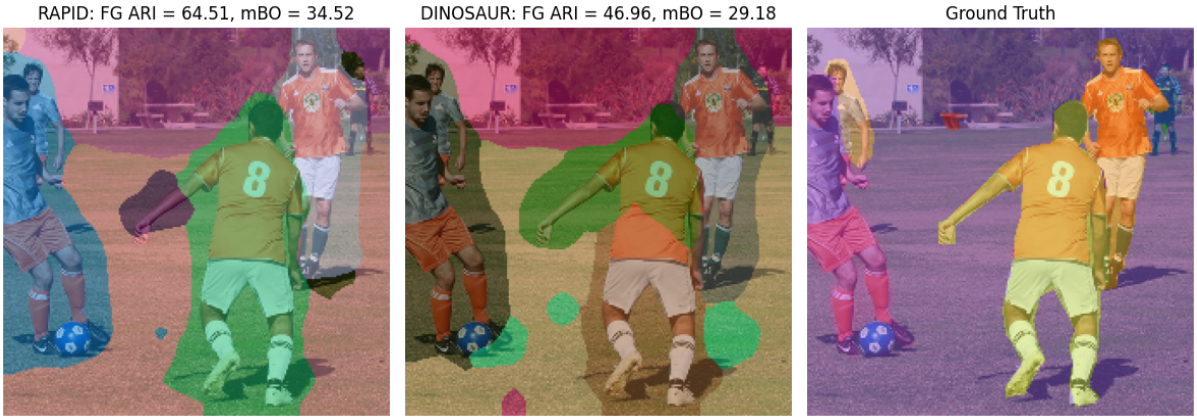


Рис. 16: Визуализация примеров из набора COCO.

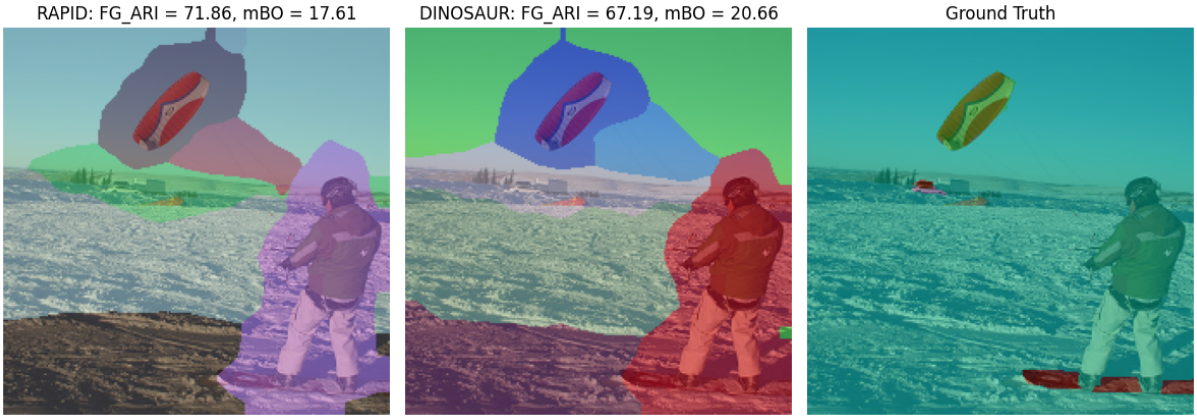


Рис. 17: Визуализация примеров из набора COCO.

5.4 Анализ эффективности модификаций

Также мы проводим всестороннюю оценку внедряемых модификаций в рамках исследования. Наши эксперименты с набором данных COCO подтверждают, что каждая из предложенных модификаций вносит существенный вклад в общую производительность и стабильность модели. Пересмотренная формулировка процедуры вывода в EM алгоритме уменьшает снижение производительности, обычно наблюдаемое в стандартном SMM. Это особенно актуально, поскольку исходный SMM основан на классической формуле гауссовой смеси, которая недостаточно использует доступную информацию во время обучения, особенно при обучении в течение ограниченного числа итераций и данных в виде мини-батчей.

Вероятностная структура нашего метода обеспечивает надежность: итеративное уточнение коэффициентов псевдовнимания и весов компонент не приводит к переобучению. Это свойство имеет решающее значение для поддержания обобщения на разных этапах обучения. Как показано в таблице 4, эти улучшения не только эффективно устраняют резкое снижение качества кластеризации, которое обычно происходит во время обучения стандартного SMM, но повышает качество представлений.

Более того, наш подход демонстрирует значительно улучшенную сходимость по сравнению с традиционной реализацией Slot Mixture. В отличие от бейзлайнов, наша модель остается стабильной на протяжении всего обучения и неизменно достигает более высоких уровней с точки зрения качества. Кроме того, как показано в таблице 5, инициализатор на основе трансформера ускоряет конвергенцию, позволяя модели достигать максимальной производительности за меньшее количество итераций.

Таблица 4: Анализ эффективности модификаций: сравнение с классическим SMM (среднее ± стандартное отклонение, 3 сида) на наборе COCO с 7-ю слотами.

	ARI	mBO	Польза
Стандартный SMM	42.5 ± 0.1	28.2 ± 0.1	–
Стандартный SMM с иниц. на основе трансф.	42.3 ± 0.2	28.1 ± 0.1	✗
Стандартный SMM с расписанием для LR	43.3 ± 0.2	28.9 ± 0.1	✓
Стандартный SMM с ИК	44.9 ± 0.3	29.3 ± 0.3	✓
Стабилизированный SMM (algorithm 1)	44.2 ± 0.2	29.7 ± 0.0	✓
RAPID (Мы)	$\mathbf{47.1 \pm 0.4}$	$\mathbf{30.5 \pm 0.1}$	✓

Таблица 5: Анализ эффективности модификация: кол-во шагов обучения, необходимое для достижения пикового качества (среднее ± стандартное отклонение, 3 сида) на наборе COCO с 7-ю слотами.

	ARI	mBO	Польза
Без инициализатора на основе трансф.	$111\text{k} \pm 34\text{k}$	$97\text{k} \pm 48\text{k}$	✗
С инициализатором на основе трансф.	$\mathbf{52k \pm 15k}$	$\mathbf{84k \pm 20k}$	✓

6 Заключение

Одним из серьёзнейших вызовов в объектно-центричном обучении моделей является их масштабирование на сложные данные реального мира. Данная работа посвящена решению этой проблемы путем стабилизации обучения объектно-центричных представлений в условиях вероятностного моделирования слотов. Основными особенностями данной модели являются:

1. Переосмысленный алгоритм обновления слотовых представлений. В частности, предлагается обновлять оценку на апостериорное распределение коэффициентов псевдовнимания после обновления каждого из ключевых параметров смеси.

2. Иерархическая конкатенация, которая представляет собой аггрегацию ключевых признаков, полученных с энкодера.

3. Новое расписание изменения значений для коэффициента скорости обучения — линейный разгон с косинусным затуханием и экспоненциальным «хвостом».

4. Параметризованный инициализатор слотовых представлений на основе трансформера, ускоряющий сходимость моделей.

Разработанная модель демонстрирует серьёзное превосходство над бейзлайнами в синтетических средах, а также имеет преимущество на сложных данных реального мира. Однако здесь стоит отметить что на наборе PASCAL VOC не удалось продемонстрировать весомого преимущества по сравнению с классическим SMM. Что можно объяснить специфичностью набора данных, который имеет достаточно серьёзные ограничения, влияющие на качество результирующих представлений.

Интригующим направлением для дальнейшего развития предлагаемого метода является его распространение на видеоданные. Однако вероятностная формулировка SMM предполагает независимость между отдельными элементами выборки, в том числе по временным шагам, что создает значительную проблему для моделирования временных последовательностей. В рамках нашей будущей работы мы планируем включить механизмы моделирования временных зависимостей в наш фреймворк, чтобы устранить это ограничение. Предварительные эксперименты в этом направлении показали многообещающие результаты, и мы планируем изучить более принципиальные подходы к временному моделированию в рамках стабилизированного SMM.

Результаты работы были представленны:

- COMPAS: Compose Actions and Slots in Object Centric World Models / V. Vorobyov et al. The Thirty-Ninth Annual Conference on Neural Information Processing Systems. - 2025 (Находится на ревью)
- RAPID: Robust Adaptive Probabilistic Inference with DINO features / V. Frolov et al. The 20th International Conference on Hybrid Artificial Intelligence Systems. - 2025 (Находится на ревью)

Список литературы

1. ImageNet: A large-scale hierarchical image database / J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei // In CVPR. — 2009.
2. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) / M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, A. Zisserman // URL http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. — 2012.
3. *Kingma D. P.*, *Ba J.* Adam: A Method for Stochastic Optimization // arXiv preprint, URL `https://arXiv.org/abs/1412.6980`. — 2014.
4. Learning phrase representations using rnn encoder–decoder for statistical machine translation / K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio // In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734. — 2014.
5. Microsoft COCO: Common Objects in Context / T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick // In ECCV. — 2014.
6. *Hinton G. E.*, *Vinyals O.*, *Dean. J.* Distilling the Knowledge in a Neural Network // In NeurIPS 2014 Deep Learning Workshop. — 2015.
7. *Walker J.*, *Gupta A.*, *Hebert M.* Dense optical flow prediction from a static image // Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2443-2451. — 2015.
8. Deep Residual Learning for Image Recognition / K. He, X. Zhang, S. Ren, J. Sun // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778. — 2016.
9. *Jang E.*, *Gu S.*, *Poole B.* Categorical Reparameterization with Gumbel-Softmax // In ICLR (Poster). — 2016.
10. Attention Is All You Need / A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin. — 2017.
11. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning / J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, R. Girshick // In CVPR. — 2017.
12. *Maddison C. J.*, *Mnih A.*, *Teh Y. W.* The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables // In ICLR (Poster). — 2017.

13. Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation / J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, J. Malik // IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(1), pp. 128-140. — 2017.

14. Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes / H. A. Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, C. Rother // International Journal of Computer Vision. — 2018.

15. *Burgess C.*, *Kim H.* 3d shapes dataset // https://github.com/deepmind/3dshapes-dataset/. — 2018.

16. Deep ordinal regression network for monocular depth estimation / H. Fu, M. Gong, C. Wang, K. Batmanghelich, D. Tao // In CVPR. — 2018.

17. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. / D. Sun, X. Yang, M.-Y. Liu, J. Kautz // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8934-8943. — 2018.

18. Shapestacks: Learning visionbased physical intuition for generalised object stacking / O. Groth, F. B. Fuchs, I. Posner, A. Vedaldi // In Proceedings of the European Conference on Computer Vision (ECCV), pp. 724–739. — 2018.

19. *Mahendran A.*, *Thewlis J.*, *Vedaldi. A.* Cross pixel optical-flow similarity for self-supervised learning // In Asian Conference on Computer Vision, pp. 99–116. Springer. — 2019.

20. Multi-Object Representation Learning with Iterative Variational Inference / K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, A. Lerchner // International conference on machine learning, PMLR, pp. 2424-2433. — 2019.

21. Spatial Broadcast Decoder: A Simple Architecture for Learning Disentangled Representations in VAEs / N. Watters, L. Matthey, C. P. Burgess, A. Lerchner // arXiv preprint arXiv:1901.07017. — 2019.

22. A survey on deep learning techniques for stereo-based depth estimation / H. Laga, L. V. Jospin, F. Boussaid, M. Bennamoun // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2020.

23. Entity abstraction in visual model-based reinforcement learning. / R. Veerapaneni, J. D. Co-Reyes, M. Chang, M. Janner, C. Finn, J. Wu, J. Tenenbaum, S. Levine // In Proceedings of the Conference on Robot Learning, PMLR 100:1439-1456. — 2020.

24. *Girdhar R.*, *Ramanan D.* CATER: A diagnostic dataset for Compositional Actions and TEmporal Reasoning // In ICLR. — 2020.

25. Object-Centric Learning with Slot Attention / F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, T. Kipf // in NeurIPS 2020. — 2020.

26. SDC-depth: Semantic divide-andconquer network for monocular depth estimation / L. Wang, J. Zhang, O. Wang, Z. Lin, H. Lu // In CVPR. — 2020.

27. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale / A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby // In ICLR. — 2021.

28. Deep learning for monocular depth estimation: A review / Y. Ming, X. Meng, C. Fan, H. Yu // Neurocomputing. — 2021.

29. Emerging Properties in Self-Supervised Vision Transformers / M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, A. Joulin // In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9650-9660. — 2021.

30. *Engelcke M.*, *Jones O. P.*, *Posner I.* GENESIS-V2: Inferring Unordered Object Representations without Iterative Refinement // In NeurIPS. — 2021.

31. *Karazija L.*, *Laina I.*, *Rupprecht C.* ClevrTex: A Texture-Rich Benchmark for Unsupervised Multi-Object Segmentation // In NeurIPS. — 2021.

32. PTR: A Benchmark for Part-based Conceptual, Relational, and Physical Reasoning / Y. Hong, L. Yi, J. Tenenbaum, A. Torralba, C. Gan // In NeurIPS. — 2021.

33. Zero-Shot Text-to-Image Generation / A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, I. Sutskever // Proceedings of the 38th International Conference on Machine Learning, PMLR. — 2021.

34. Conditional Object-Centric Learning from Video / T. Kipf, G. F. Elsayed, A. Mahendran, A. Stone, S. Sabour, G. Heigold, R. Jonschkowski, A. Dosovitskiy, K. Greff // In ICLR 2022: The Tenth International Conference on Learning Representations. — 2022.

35. Kubric: A Scalable Dataset Generator / K. Greff [и др.] // In CVPR. — 2022.

36. SAVi++: Towards End-to-End Object-Centric Learning from Real-World Videos / G. F. Elsayed, A. Mahendran, S. van Steenkiste, K. Greff, M. C. Mozer, T. Kipf // In NeurIPS. — 2022.

37. *Singh G.*, *Deng F.*, *Ahn S.* Illiterate DALL-E Learns to Compose // In ICLR 2022: The Tenth International Conference on Learning Representations. — 2022.

38. *Singh G.*, *Wu Y.-F.*, *Ahn S.* Simple Unsupervised Object-Centric Learning for Complex and Naturalistic Videos // In NeurIPS. — 2022.

39. Bridging the Gap to Real-World Object-Centric Learning / M. Seitzer, M. Horn, A. Zadaianchuk, D. Zietlow, T. Xiao, C.-J. Simon-Gabriel, T. He, Z. Zhang, B. Schölkopf, T. Brox, F. Locatello // In ICLR 2023: The Eleventh International Conference on Learning Representations. — 2023.

40. Shepherding Slots to Objects: Towards Stable and Robust Object-Centric Learning / J. Kim, J. Choi, H.-J. Choi, S. J. Kim // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 19198-19207. — 2023.

41. Identifiable Object-Centric Representation Learning via Probabilistic Slot Attention / A. Kori, F. Locatello, A. Santhirasekaram, F. Toni, B. Glocker, F. Ribeiro // Advances in Neural Information Processing Systems. — 2024.

42. Object-Centric Learning with Slot Mixture Module / D. Kirilenko, V. Vorobyov, A. Kovalev, A. Panov // In ICLR. — 2024.