The 10th International Conference on
Deep Learning in Computational Physics

July 7, 2026, Moscow, Russia

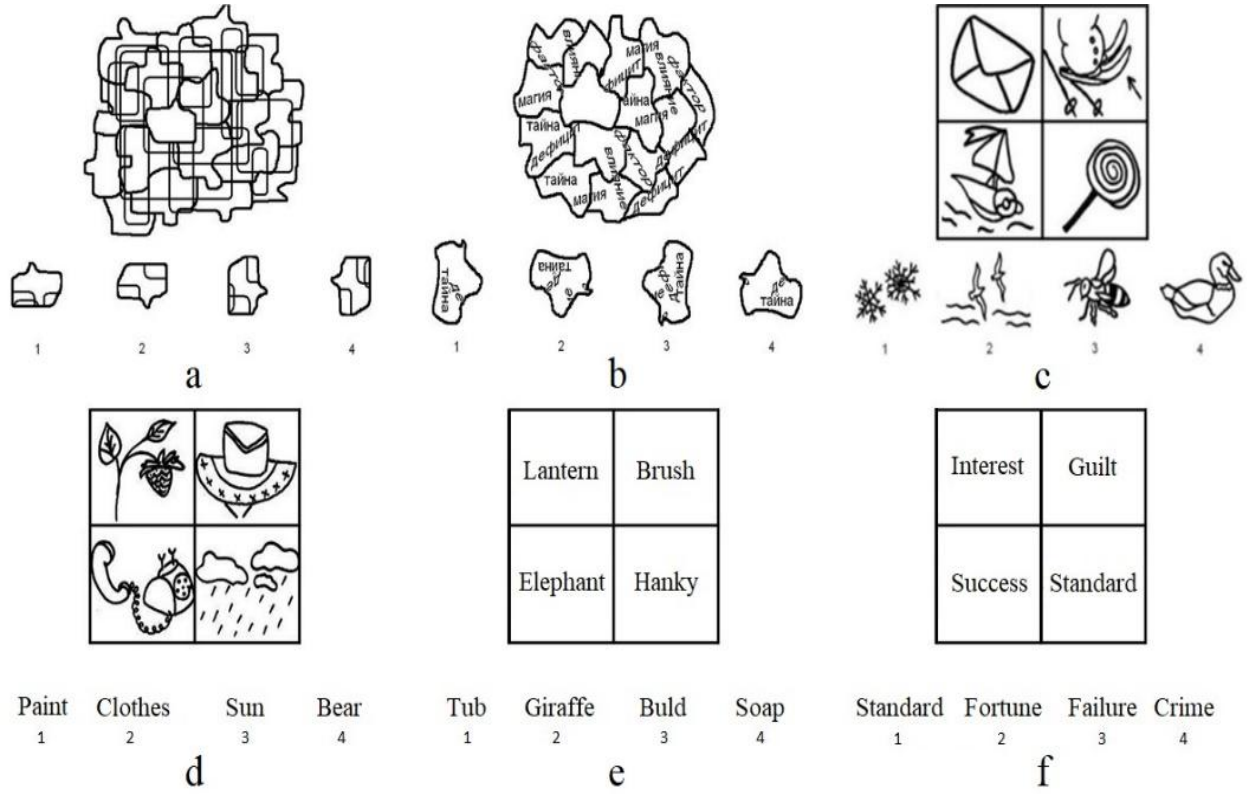
Сравнение глубоких архитектур при различных уровнях предобработки фМРТ в задаче бинарной классификации мозговой активности

Макаров Александр Сергеевич[1], Гаджиев Исмаил Маратович[1],
Доленко Сергей Анатольевич[2]

[1] Физический факультет, Московский государственный университет имени М.В.Ломоносова

[2] Научно-исследовательский институт ядерной физики имени Д.В. Скобельцына, Московский государственный университет имени М.В.Ломоносова

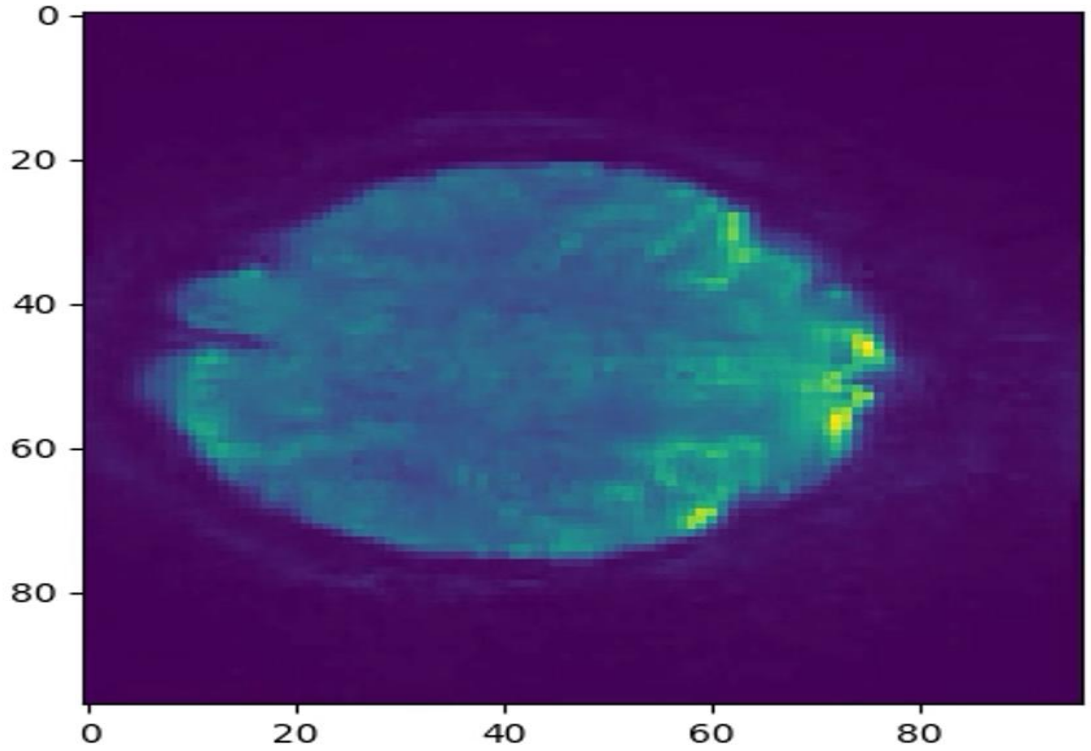
Экспериментальная парадигма



- 31 испытуемый;
- Задания 6 типов;
- Каждому заданию соответствует свое когнитивное состояние;
- синхронно записывалась фМРТ.

Идея: по паттернам фМРТ понять, какую задачу решает человек в данный момент.

Данные фМРТ



- Количество временных отсчетов для каждого испытуемого: 3620;
- Размерность изображения мозга: 91x109x91.
- Объем памяти для записи сеанса одного испытуемого: 26 Гб

Нетревиально: огромные данные, избыточность, шум, артефакты движения.

Предыдущие исследования: классические методы

- Подход к решению:

  предобработка -> АГК -> Вычитание -> погружение -> ML-модель;

- Протестированы:

  Логистическая регрессия(LR) и градиентный бустинг(LGBM);

- Лучший результат:

  LGBM с учетом предыстории в **25 шагов** | **ROC-AUC = 0.894**

Текущие исследования: переход к глубокому обучению

- Заменить классический методы сквозным глубоким обучением;
- Сравнить глубокие модели с результатами классичиских моделей;
- Проверить, способны ли глубокие модели извлекать полезные признаки напрямую из сырых и минимально обработанных фМРТ данных.

Типы входных данных

- RawData:

  полностью сырые дынные (максимум информации, но и максимум шума);

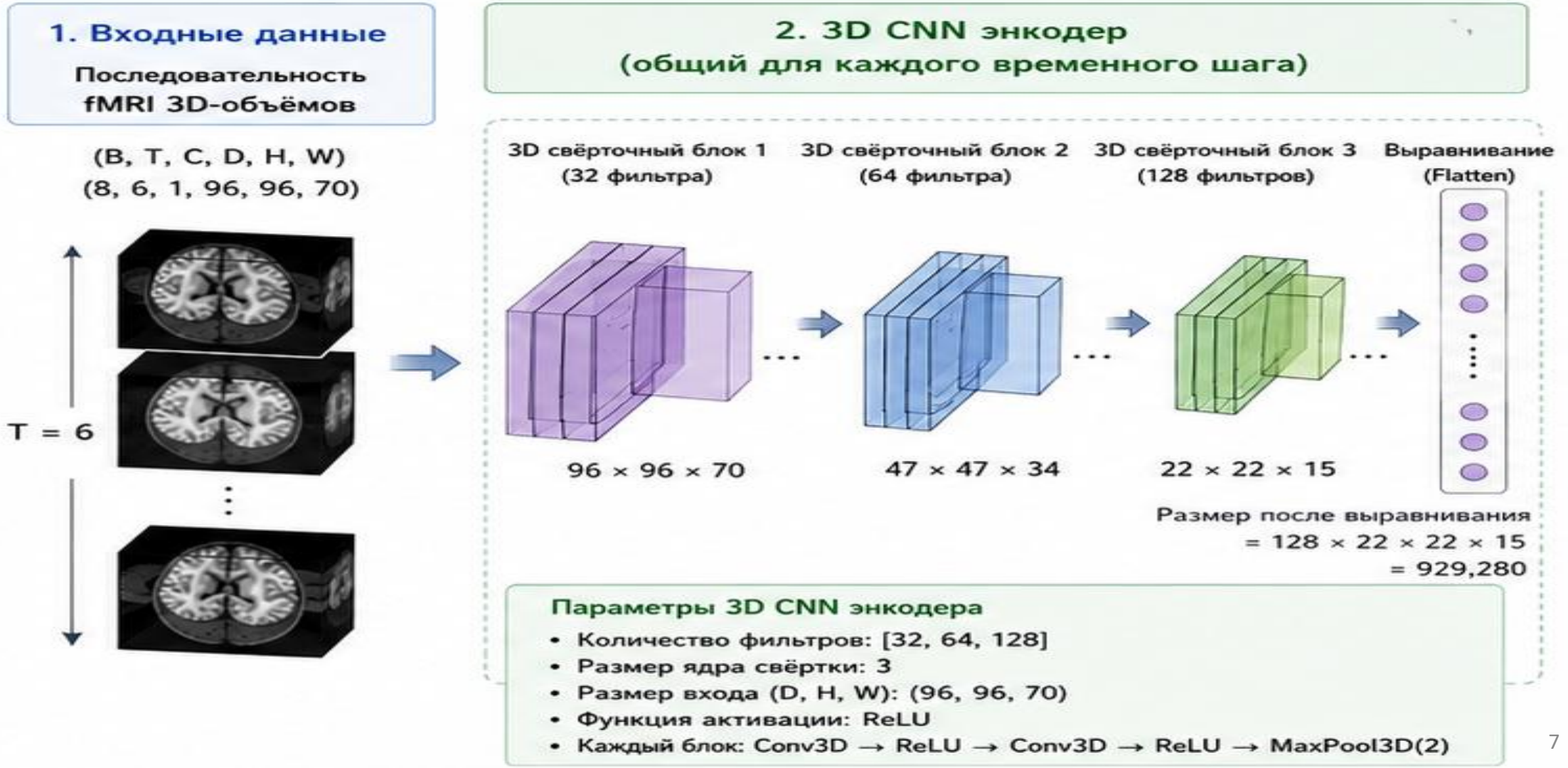
- MCData:

  данные с коррекции артефактов движения;

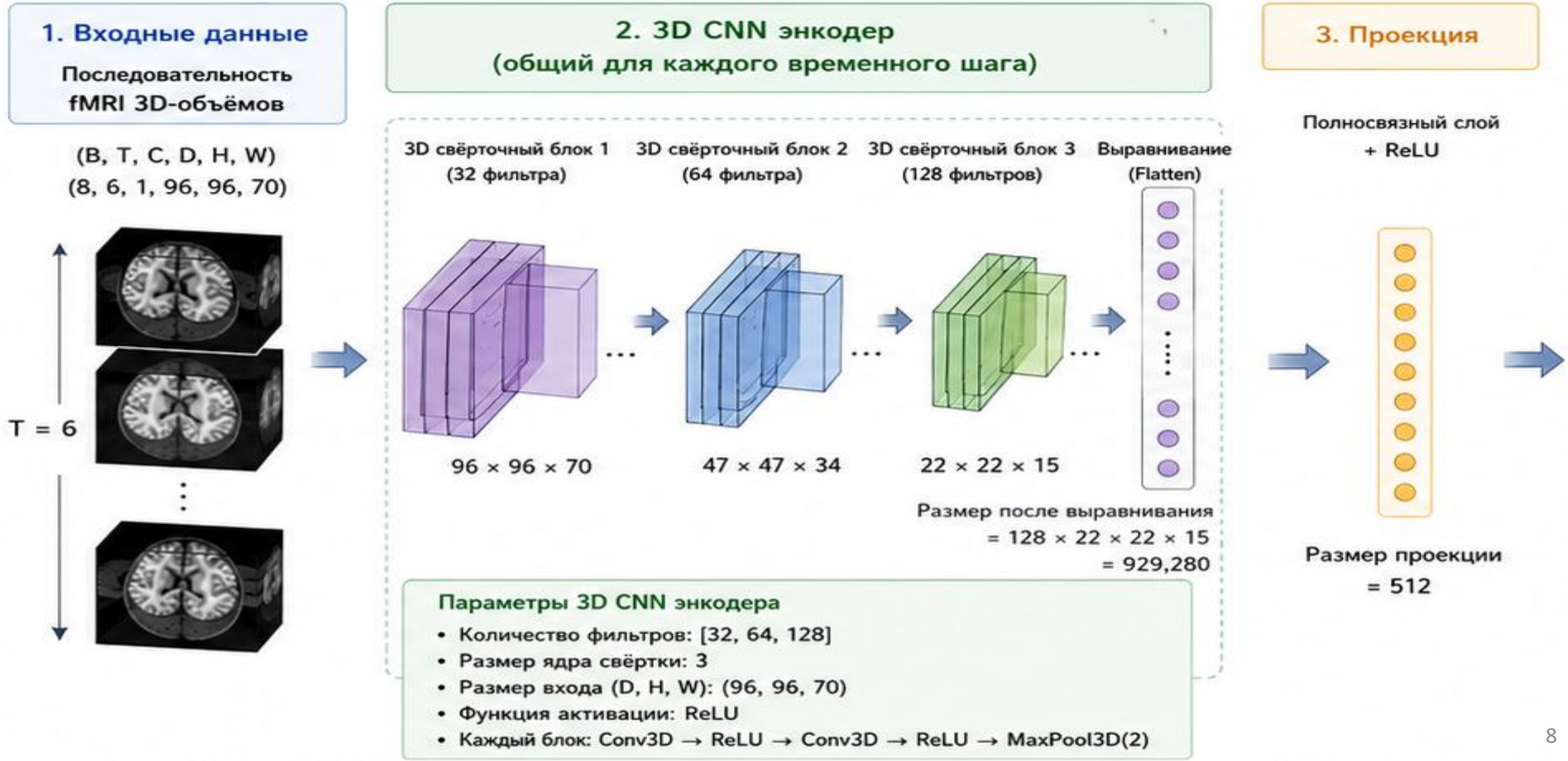
- FullData:

  данные с полным циклом предобработки.

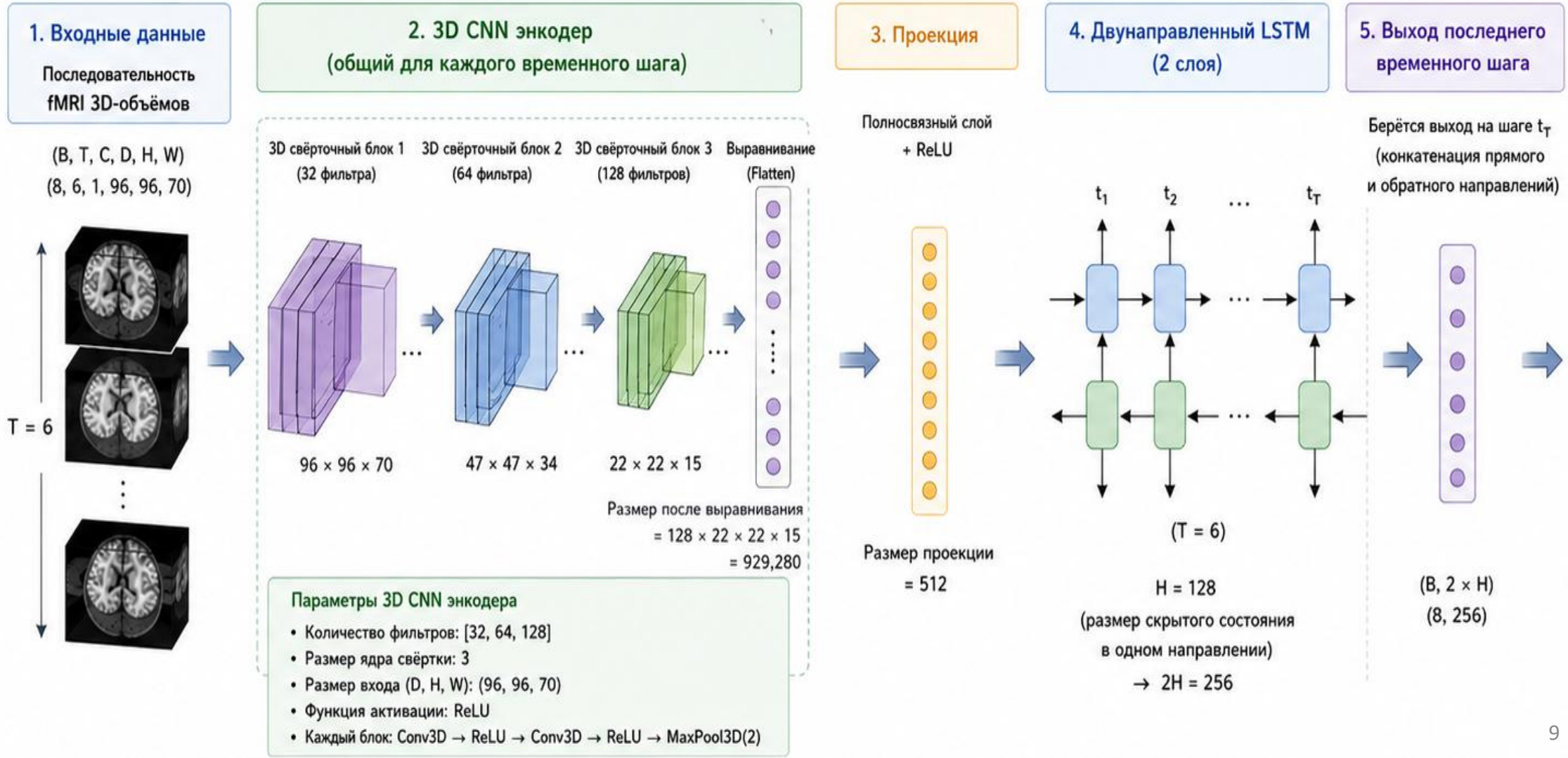
Архитектура модели: CNNLSTM



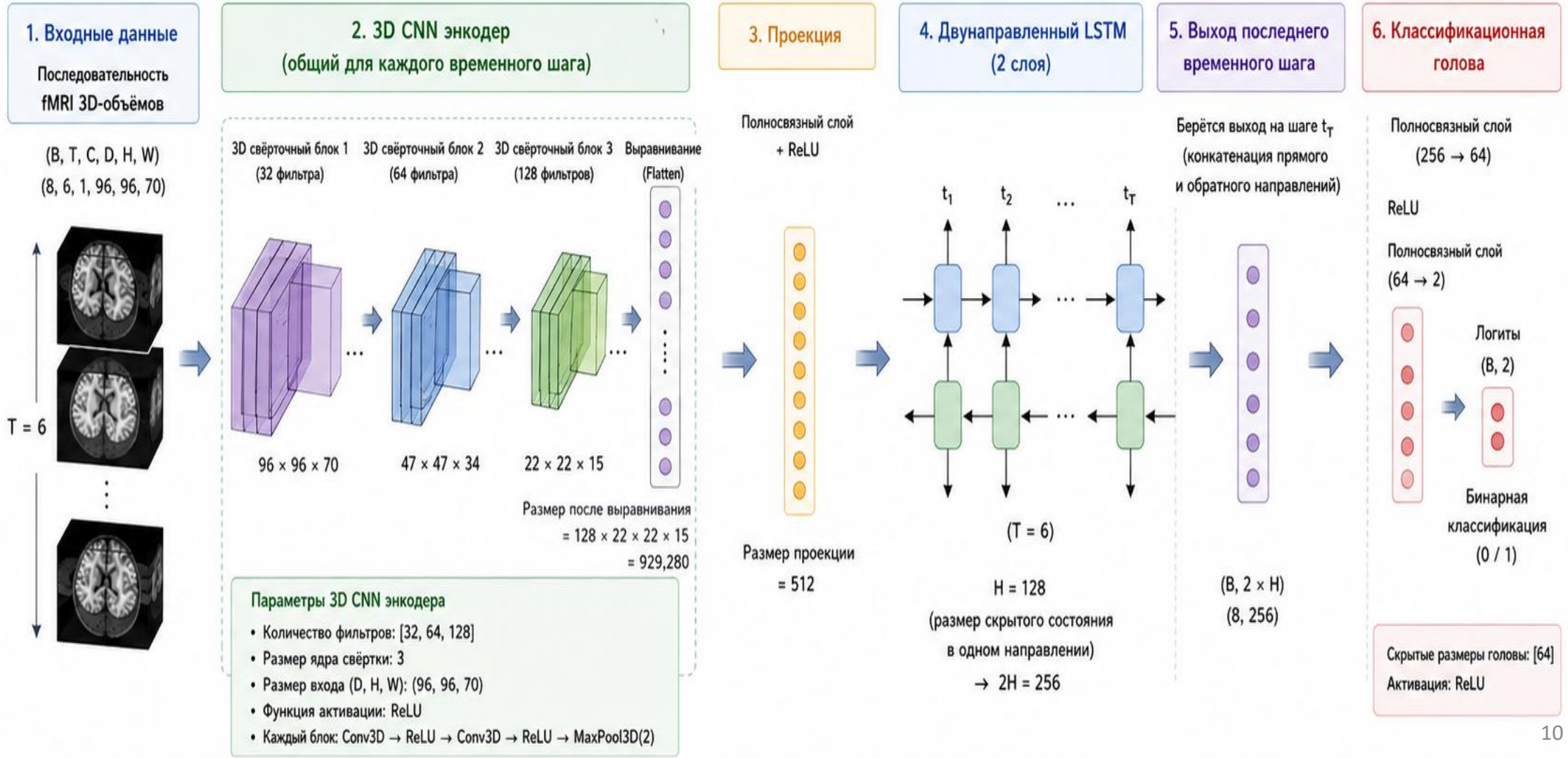
Архитектура модели: CNNLSTM



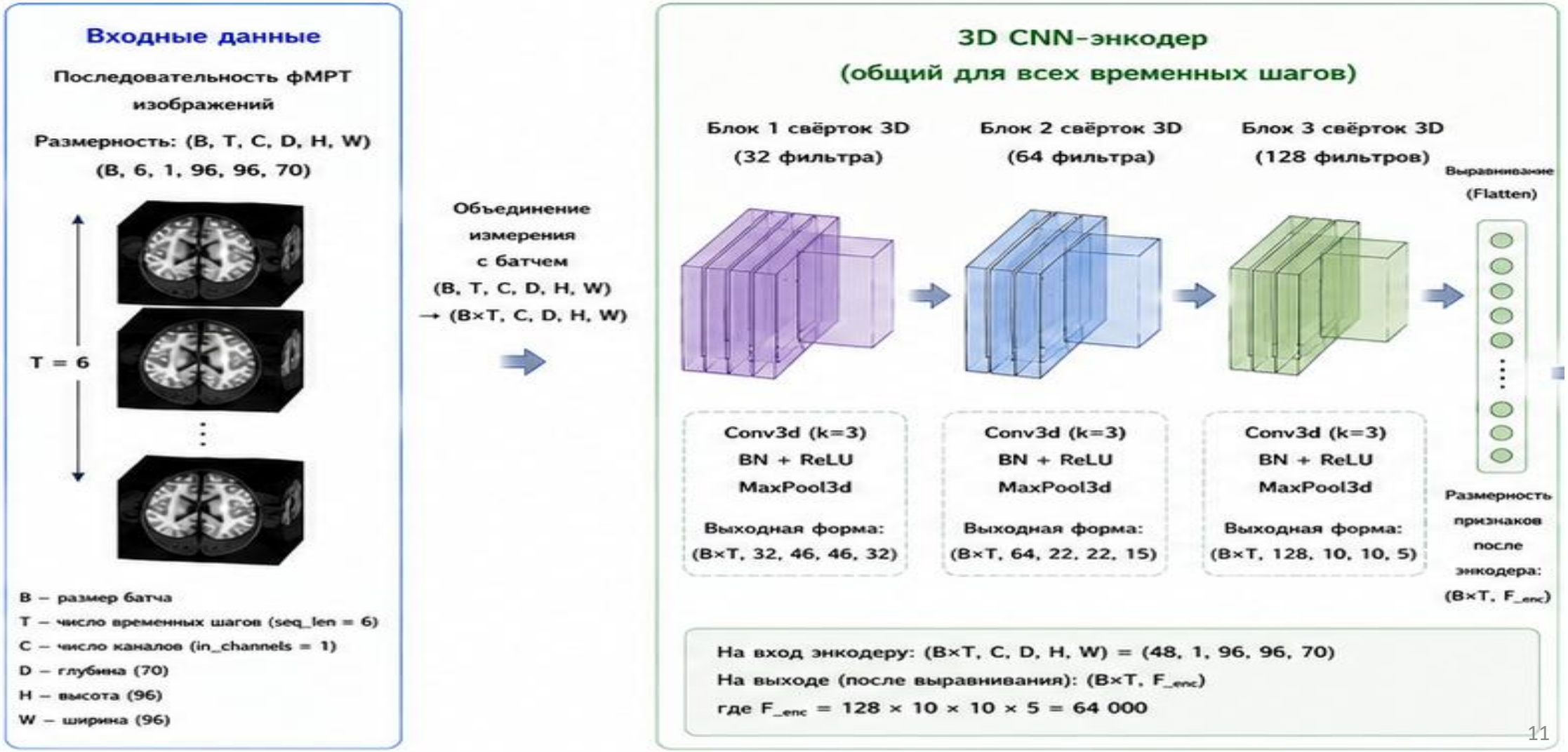
Архитектура модели: CNNLSTM



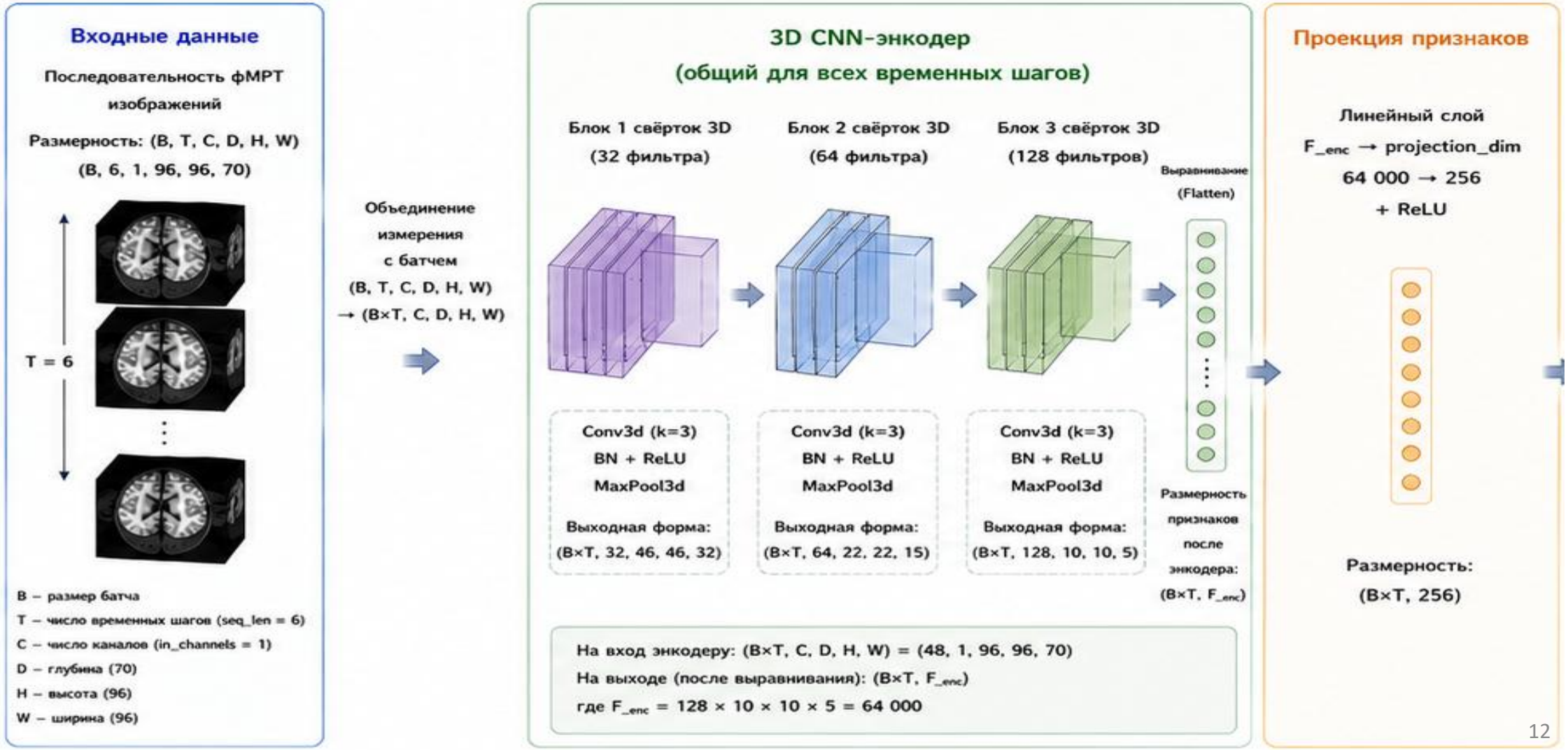
Архитектура модели: CNNLSTM



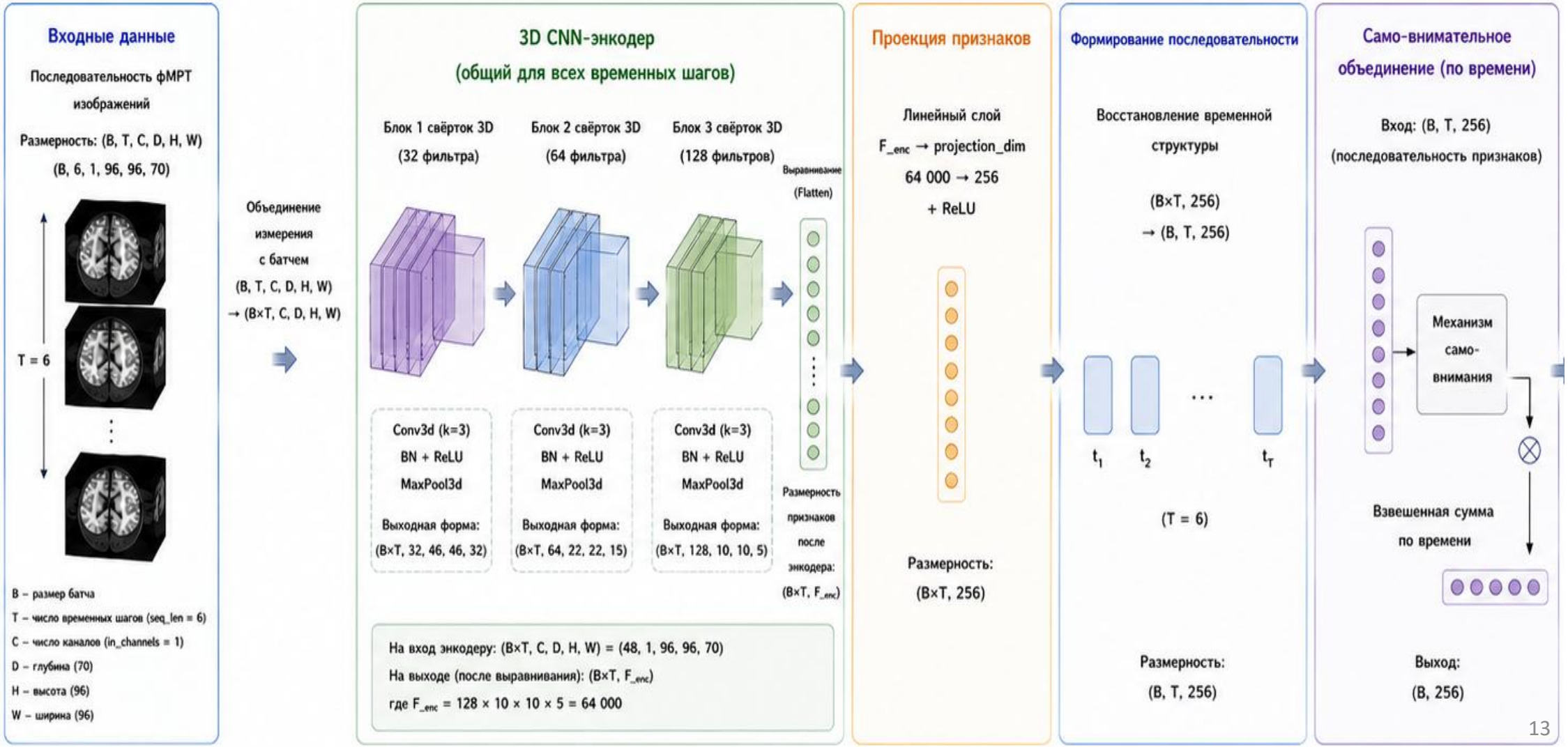
Архитектура модели: CNNAttention



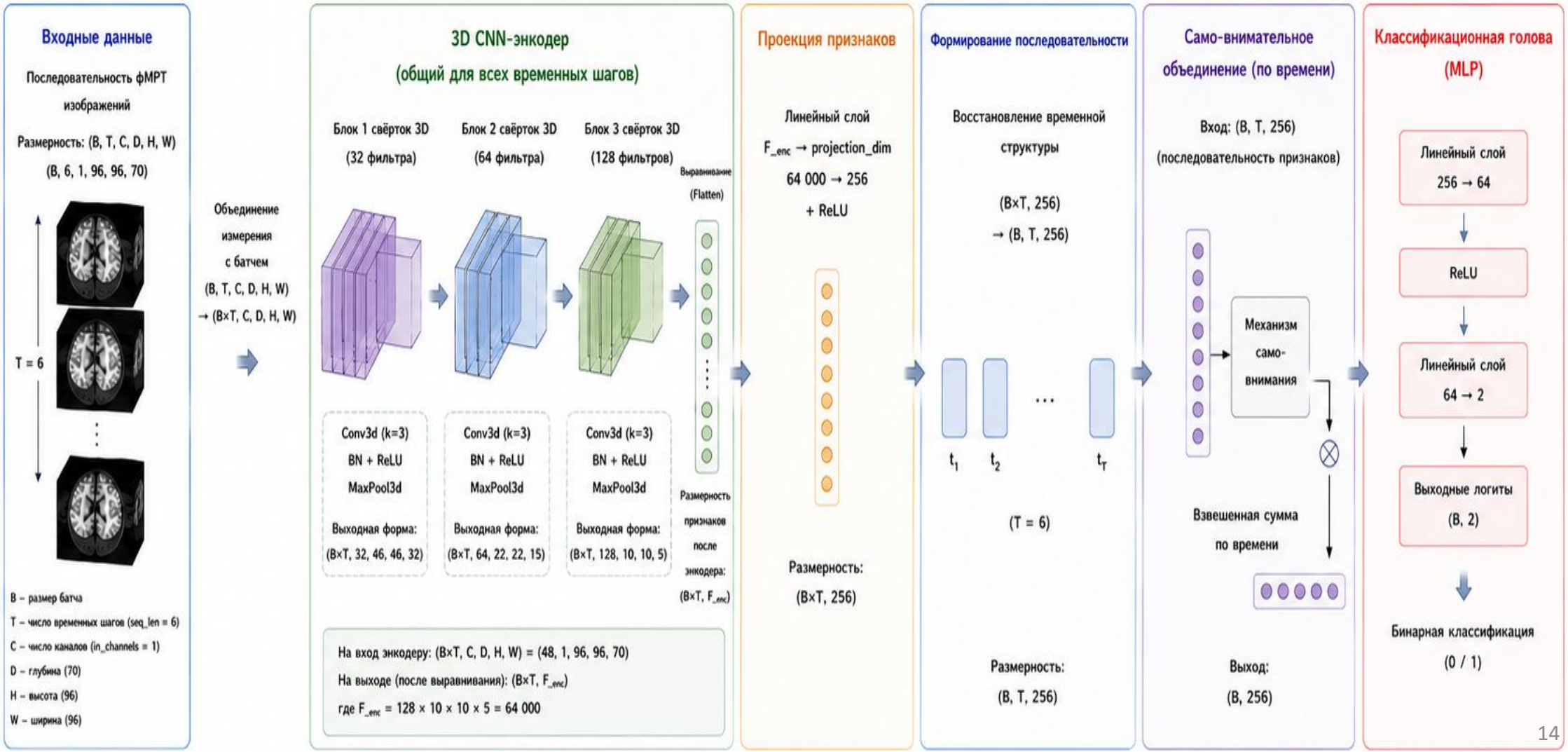
Архитектура модели: CNNAttention



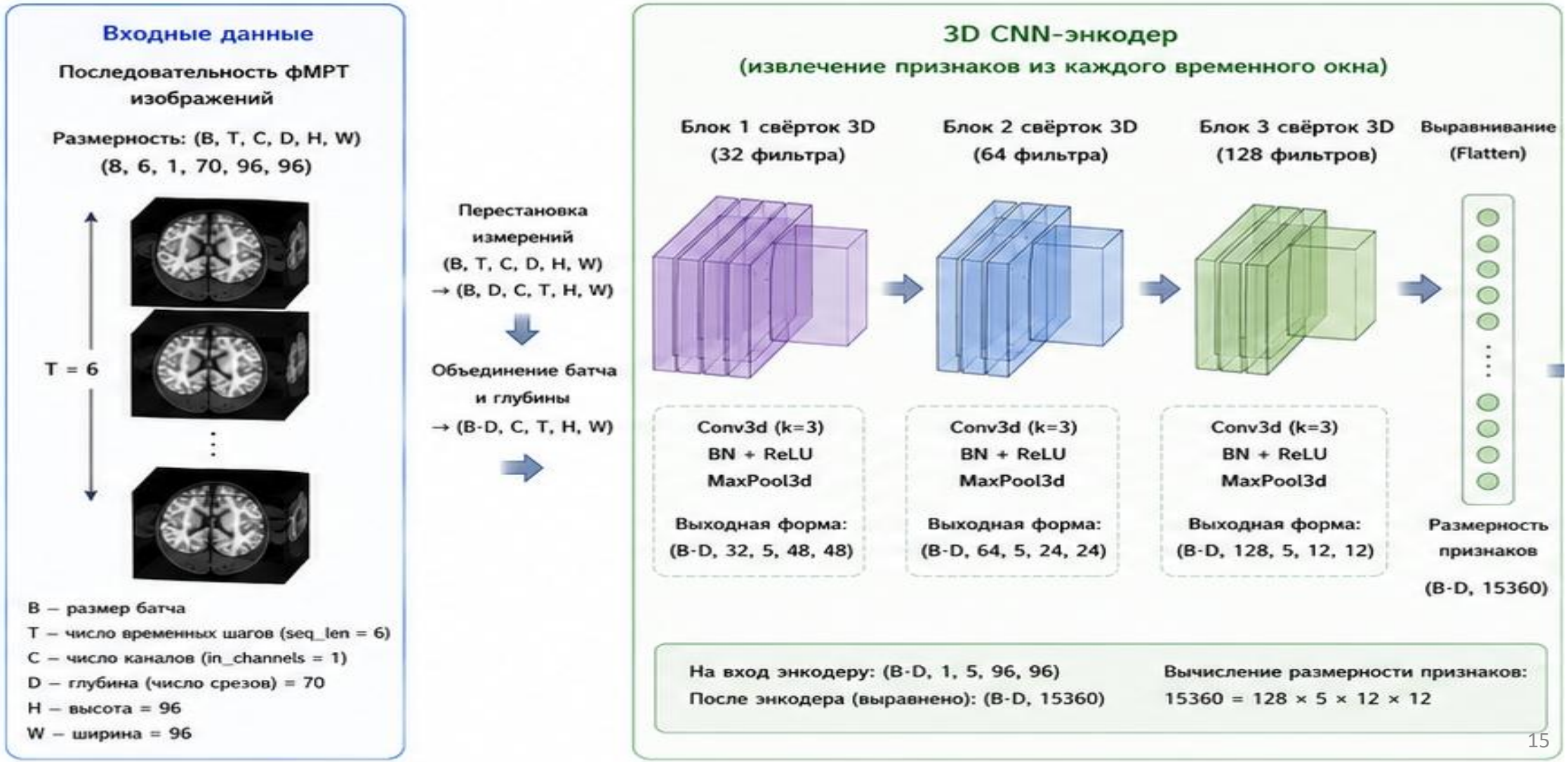
Архитектура модели: CNNAttention



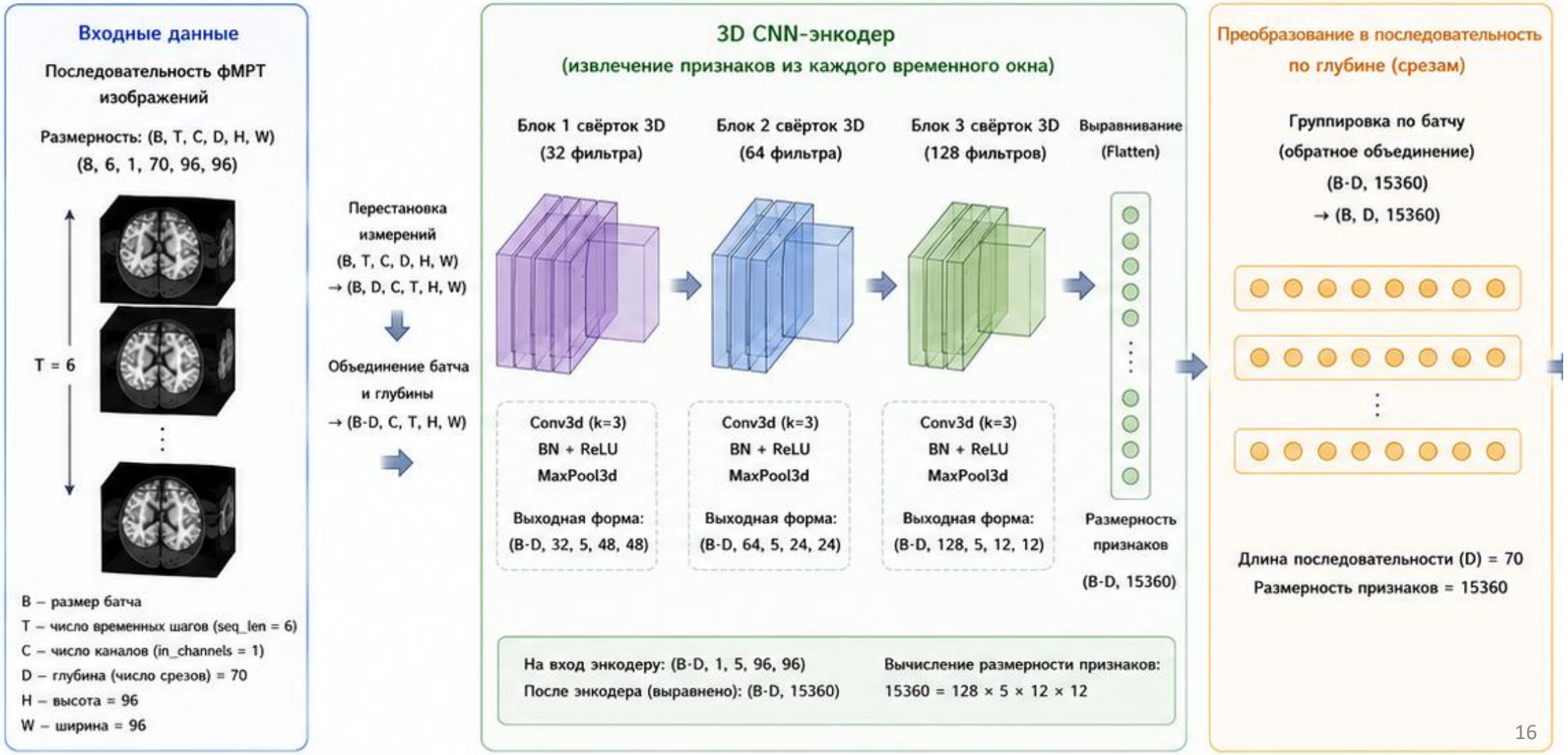
Архитектура модели: CNNAttention



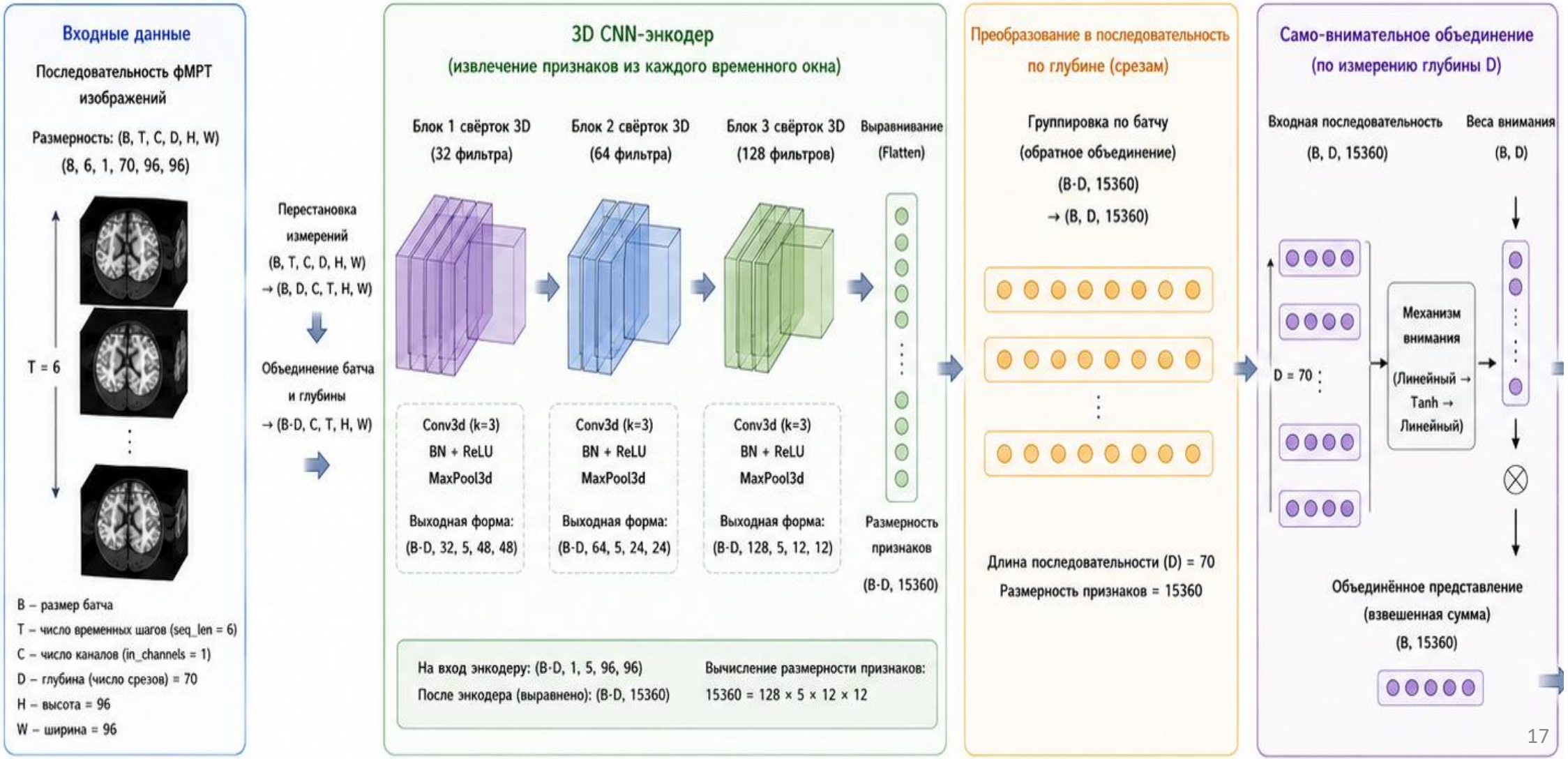
Архитектура модели: AttentionZ



Архитектура модели: AttentionZ

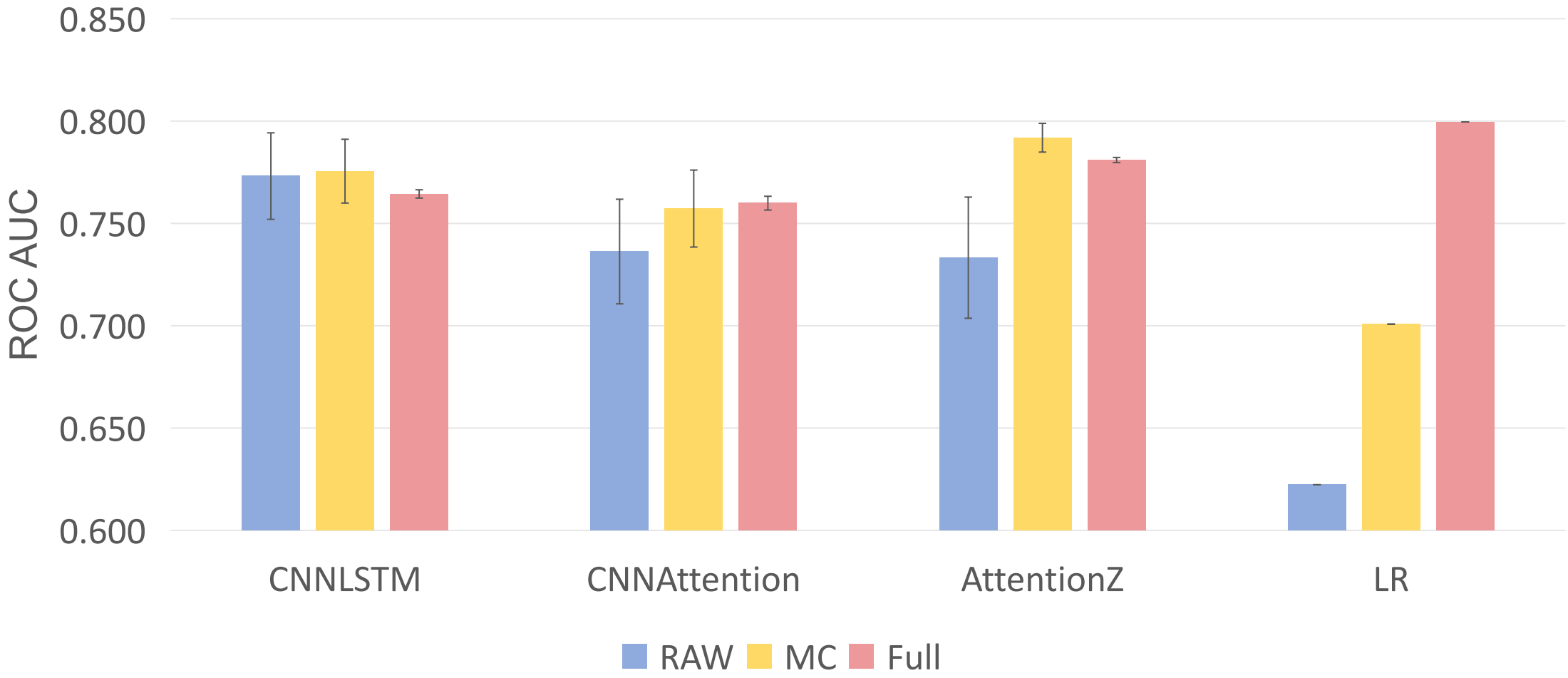


Архитектура модели: AttentionZ



Архитектура модели: AttentionZ

Результаты классификации



Выводы

- Логистическая регрессия показывает лучший результат
  на полном цикле предобработки (Full);

- Для глубоких сетей результат на полном конвейере (Full)
  либо сравним с результатами на данных с коррекцией движения (MC),
  либо в ряде случаев уступает ему;

- Результаты на сырых данных показали качество либо сравнимое,
  либо хуже, чем на данных с предобработкой.

Особенности

- Эксперименты проводились на сравнительно маленьких последовательностях (5 изображений): при увеличении длины, ситуация между классическими моделями и нейросетями может измениться;

- Не проверены промежуточные комбинации этапов предобработки: возможно существует «золотая середина» для глубоких сетей;

- Решалась только задача бинарной классификации: возможно при многоклассовой классификации результаты окажутся иными.

Дальнейшие исследования

- Продолжить тестирование и улучшение архитектур моделей;
- Провести эксперименты с увеличением длины входной последовательности изображений, чтобы захватить более глубокие пространственно-временные зависимости;
- Провести эксперименты с различными комбинациями этапов предобработки;
- Провести экспериметы с предобученными моделями;
- Перейти к решению задачи многоклассовой классификации.

Спасибо за внимание!

Почему мы не использовали предобученные модели?

1. Каждая задача требует своего подхода:

- Разные данные, разные пространственные размерности, разная предобработка;
- Воспроизвести чужую модель на своих данных часто не так просто, как кажется;
- Сперва нужно иметь собственные бейзлайны для сравнения.

2. Предобученные модели - следующий шаг:

- Например, NeuroSTORM позволяет заморозить все веса и дообучать только голову и входные токены - это следующее, что мы хотим попробовать.

Масштаб экспериментов

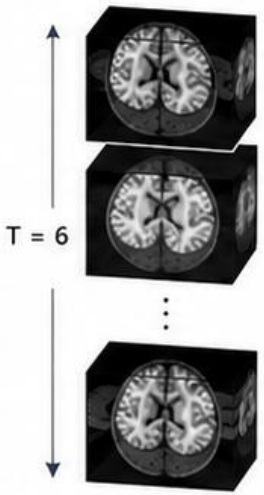
- Для подбора гиперпараметров было проведено порядка 50 запусков;
- Каждая модель обучалось в среднем 10 часов;
- Суммарно примерно 500 часов чистого обучения;
- Дополнительно тестировались и другие архитектуры, но они показали результаты хуже представленных.

Архитектура модели LSTMCNN: 3D CNN энкодер + двунаправленный LSTM + классификационная голова

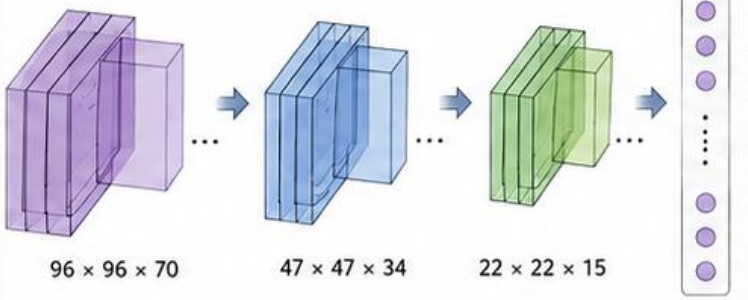
1. Входные данные

Последовательность fMRI 3D-объёмов

(B, T, C, D, H, W)
(8, 6, 1, 96, 96, 70)



2. 3D CNN энкодер (общий для каждого временного шага)



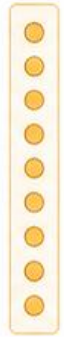
Размер после выравнивания
= 128 × 22 × 22 × 15
= 929,280

Параметры 3D CNN энкодера

- Количество фильтров: [32, 64, 128]
- Размер ядра свёртки: 3
- Размер входа (D, H, W): (96, 96, 70)
- Функция активации: ReLU
- Каждый блок: Conv3D → ReLU → Conv3D → ReLU → MaxPool3D(2)

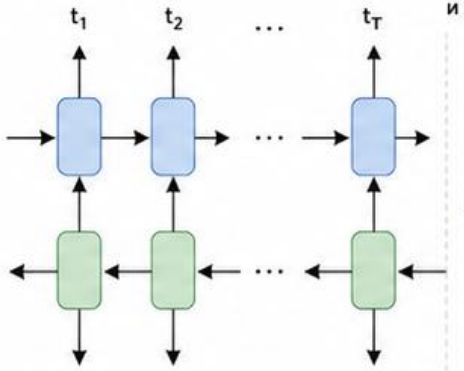
3. Проекция

Полносвязный слой
+ ReLU



Размер проекции
= 512

4. Двунаправленный LSTM (2 слоя)



(T = 6)

H = 128
(размер скрытого состояния
в одном направлении)
→ 2H = 256

5. Выход последнего временного шага

Берётся выход на шаге t_T
(конкатенация прямого
и обратного направлений)

(B, 2 × H)
(8, 256)

6. Классификационная голова

Полносвязный слой
(256 → 64)

ReLU

Полносвязный слой
(64 → 2)

Логиты
(B, 2)

Бинарная
классификация
(0 / 1)

Скрытые размеры головы: [64]
Активация: ReLU

Данные / Датасет

- Датасет: DatasetSequence
- Длина последовательности (T): 6
- Количество каналов (C): 1
- Размер входа (D, H, W): (96, 96, 70)
- Размер батча: 8

Гиперпараметры модели

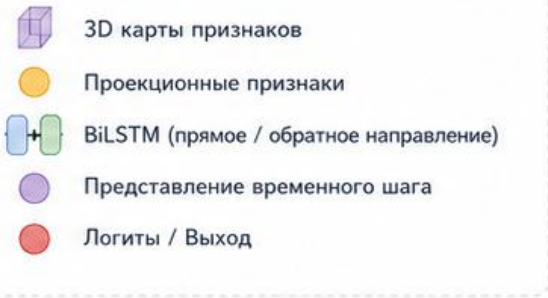
- Количество фильтров: [32, 64, 128]
- Размер проекции: 512
- Скрытое состояние LSTM: 128
- Количество слоёв LSTM: 2 (двунаправленный)
- Скрытые размеры головы: [64]

Настройки обучения

- Тип классификации: бинарная
- Функция потерь: CrossEntropyLoss
- Оптимизатор: Adam (lr = 1e-5)
- Планировщик: ReduceLROnPlateau (мониторинг: val_roc_auc, режим: max, фактор: 0.1, терпение: 1)
- Количество эпох: 7

Предобработка / Прочее

- Использовать разности (diff): да
- Среднее значение (mean): 0.0
- Стандартное отклонение (std): 8.521
- Доля валидации (val_frac): 0.15
- Случайное состояние (random_state): 1000



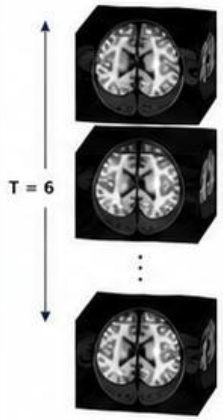
SimpleAttentionCNNModel: 3D CNN-энкодер + само-внимательное объединение + классификационная голова

Входные данные

Последовательность фМРТ изображений

Размерность: (B, T, C, D, H, W)

(B, 6, 1, 96, 96, 70)



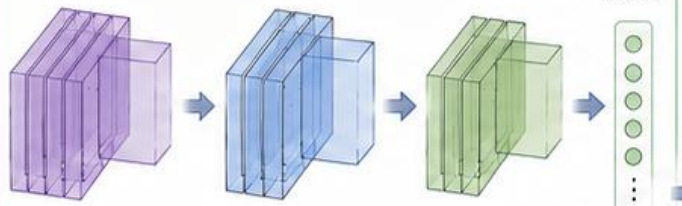
B – размер батча
T – число временных шагов (seq_len = 6)
C – число каналов (in_channels = 1)
D – глубина (70)
H – высота (96)
W – ширина (96)

Объединение измерения с батчем

(B, T, C, D, H, W) → (B×T, C, D, H, W)

3D CNN-энкодер (общий для всех временных шагов)

Блок 1 свёрток 3D (32 фильтра)	Блок 2 свёрток 3D (64 фильтра)	Блок 3 свёрток 3D (128 фильтров)
Conv3d (k=3) BN + ReLU MaxPool3d	Conv3d (k=3) BN + ReLU MaxPool3d	Conv3d (k=3) BN + ReLU MaxPool3d
Выходная форма: (B×T, 32, 46, 46, 32)	Выходная форма: (B×T, 64, 22, 22, 15)	Выходная форма: (B×T, 128, 10, 10, 5)



Выравнивание (Flatten)

Размерность признаков после энкодера: (B×T, F_{enc})

На вход энкодеру: (B×T, C, D, H, W) = (48, 1, 96, 96, 70)
На выходе (после выравнивания): (B×T, F_{enc})
где F_{enc} = 128 × 10 × 10 × 5 = 64 000

Проекция признаков

Линейный слой F_{enc} → projection_dim

64 000 → 256

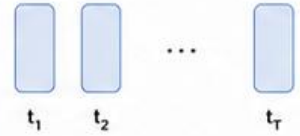
+ ReLU

Размерность: (B×T, 256)

Формирование последовательности

Восстановление временной структуры

(B×T, 256) → (B, T, 256)

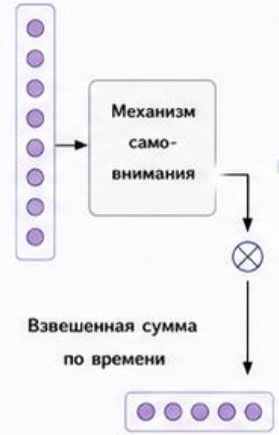


(T = 6)

Размерность: (B, T, 256)

Само-внимательное объединение (по времени)

Вход: (B, T, 256) (последовательность признаков)



Взвешенная сумма по времени

Выход: (B, 256)

Классификационная голова (MLP)

Линейный слой 256 → 64

ReLU

Линейный слой 64 → 2

Выходные логиты (B, 2)

Бинарная классификация (0 / 1)

Данные / датасет

- Датасет: DatasetSequence
- seq_len (T): 6
- in_channels (C): 1
- img_size (D, H, W): (96, 96, 70)
- batch_size: 8

Гиперпараметры модели

- num_filters: [32, 64, 128]
- kernel_size: 3
- projection_dim: 256
- attn_hidden (скрытый размер внимания): 128
- head_hidden_dims: [64]
- n_classes: 2 (бинарная классификация)
- pos_bool: false (позиционное кодирование не используется)

Параметры само-внимания

- feature_dim (на вход): 256
- hidden_dim (attn_hidden): 128
- Механизм: линейный слой → Tanh → линейный слой → softmax по времени
- Выход: взвешенная сумма признаков по временной оси

Параметры обучения

- Функция потерь: CrossEntropyLoss
- Оптимизатор: Adam (lr = 1e−6)
- Планировщик: ReduceLROnPlateau (monitor: val_roc_auc, mode: max, factor: 0.1, patience: 1)
- Эпохи: 7
- Размер батча: 8
- Доля валидации: 0.15
- Gradient clipping: не используется
- random_state: 1000

Нормализация / предобработка

- mean: 0.0
- std: 8.521
- diff: true

- 3D карты признаков (свёртки)
- Выравненные признаки (flatten)
- Проекция признаков
- Элементы последовательности (по временным шагам)
- Веса внимания / объединённые признаки
- Выходные логиты

Примечание: B – размер батча, T – число временных шагов, C – число каналов, D – глубина, H – высота, W – ширина, F_enc – размерность признаков после 3D CNN-энкодера.

AttentionZModel: 3D CNN-энкодер + само-внимательное объединение + классификационная голова

Входные данные

Последовательность фМРТ изображений

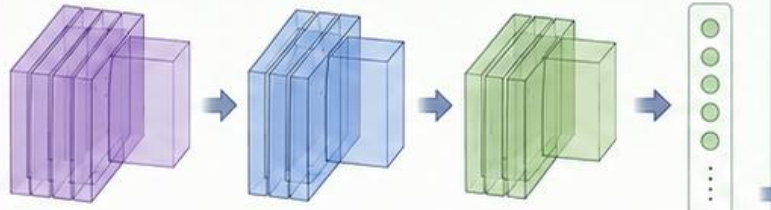
Размерность: (B, T, C, D, H, W)
(8, 6, 1, 70, 96, 96)

B – размер батча
T – число временных шагов (seq_len = 6)
C – число каналов (in_channels = 1)
D – глубина (число срезов) = 70
H – высота = 96
W – ширина = 96

Перестановка измерений
(B, T, C, D, H, W)
→ (B, D, C, T, H, W)

Объединение батча и глубины
→ (B·D, C, T, H, W)

3D CNN-энкодер
(извлечение признаков из каждого временного окна)

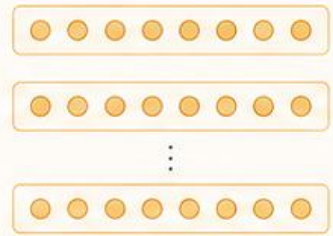


Блок 1 свёрток 3D (32 фильтра)	Блок 2 свёрток 3D (64 фильтра)	Блок 3 свёрток 3D (128 фильтров)	Выравнивание (Flatten)
Conv3d (k=3) BN + ReLU MaxPool3d	Conv3d (k=3) BN + ReLU MaxPool3d	Conv3d (k=3) BN + ReLU MaxPool3d	Размерность признаков (B·D, 15360)
Выходная форма: (B·D, 32, 5, 48, 48)	Выходная форма: (B·D, 64, 5, 24, 24)	Выходная форма: (B·D, 128, 5, 12, 12)	

На вход энкодеру: (B·D, 1, 5, 96, 96)
После энкодера (выравнено): (B·D, 15360)
Вычисление размерности признаков:
15360 = 128 × 5 × 12 × 12

Преобразование в последовательность по глубине (срезам)

Группировка по батчу
(обратное объединение)
(B·D, 15360)
→ (B, D, 15360)

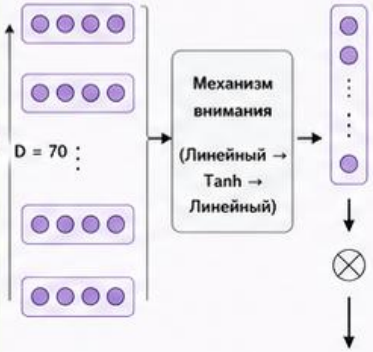


Длина последовательности (D) = 70
Размерность признаков = 15360

Само-внимательное объединение (по измерению глубины D)

Входная последовательность
(B, D, 15360)

Веса внимания
(B, D)



Объединённое представление
(взвешенная сумма)
(B, 15360)

Классификационная голова (MLP)

Бинарная классификация
(0 / 1)

Данные / датасет

- Датасет: DatasetSequence
- seq_len (T): 6
- in_channels (C): 1
- img_size (D, H, W): (70, 96, 96)
- batch_size: 8

Гиперпараметры модели

- num_filters: [32, 64, 128]
- kernel_size: 3
- attn_hidden (скрытый размер внимания): 256
- head_hidden_dims: [128]
- n_classes: 2 (бинарная классификация)
- Размерность признаков после энкодера: 15360 (128 × 5 × 12 × 12)

Параметры обучения

- Функция потерь: CrossEntropyLoss
- Оптимизатор: Adam (lr = 1e−4)
- Планировщик: ReduceLROnPlateau (monitor: val_roc_auc, mode: max, factor: 0.1, patience: 2)
- Эпох: 7
- Размер батча: 8
- Доля валидации: 0.15
- Gradient clipping: не используется
- random_state: 1000

Нормализация / предобработка

- mean: 0.0
- std: 8.521
- diff: true

- 3D карты признаков (после свёрток)
- Выравненные признаки
- Элементы последовательности (по срезам глубины)
- Веса внимания / объединённые признаки
- Выходные логиты

Примечание: B – размер батча, T – число временных шагов, C – число каналов, D – глубина (срезы), H – высота, W – ширина

Введение

Динамический Когновизор – инструмент, который на основе анализа данных о мозговой активности распознает и визуализирует когнитивные состояния и переходы между ними.

- *Мозговая активность* фиксируется с помощью функциональной магнитно-резонансной томографии (фМРТ);
- *Когнитивное состояние* – состояние, в котором задействуется один или несколько типов мышления.