Known Issues

????????? ???????? ? ??????????? ? ????????????

Список вещей, которые не работают как задокументировано в других местах репозитория (корневой `README.md`, `make_docs/`), или которые могут удивить при первом использовании. Составлен по фактическому чтению кода на момент написания этой документации.

????

`ComponentsObject.update_wear()` / `.predict_fix()` ?????? ? `AttributeError`

`src/mlumina/components_wear/components_object.py:163` и `:272` вызывают несуществующий метод `self._get_wear_dict()`. Реально определён `_get__dict()` (строка 74, с двойным подчёркиванием). Воспроизводится в штатном примере `examples/components_wear/example.py:33`. Подробности и обходной путь — components-wear.md.

????????? ????????? `README.md` ? ?????

Примеры кода в разделе "Использования фреймворка MLumina" корневого `README.md` местами не соответствуют реальному API:

В README	В коде
`from mlumina import hello; hello()`	`src/mlumina/__init__.py` пуст, функции `hello` не существует
`YoloImageModel(model_path="yolov8n.pt")`	`YoloImageModel(model)` — принимает готовый объект `ultralytics.YOLO`, не путь

В README	В коде
`model.predict_parallel(image_paths=..., batch_size=...)`	метод называется `predict_ray(images, num_workers=2, **kwargs)`
`from mlumina.interpolation_tables import InterpolationTable, evaluate_models_parallel`	класс называется `TableInterpolator`; функции `evaluate_models_parallel` нет — ближайшие аналоги `compare_multiple_models`/`compare_multiple_models_ray`
`graph.calculate_parallel()`	метода нет; есть `simulate_objects`, `simulate_objects_ray`, `simulate_objects_bn`

Раздел README про XGBoost (`xgboost_load_dataset`, `XGBoostModel`, `XGBoostTuningModel`) — единственный, полностью совпадающий с кодом.

`make_docs/` (Sphinx) ?? ???????? ??????? ???

`make_docs/source/introduction/what_is_mlumina.rst` описывает MLumina как открытый мультимодальный AutoML-фреймворк с эволюционным поиском пайплайнов, поддержкой классификации/регрессии/временных рядов и сохранением/загрузкой найденных решений. В коде ничего из этого нет — библиотека представляет собой набор независимых обёрток и утилит (см. README). Единственный фрагмент кода в этом файле (`TableInterpolator`) — рабочий и совпадает с реальным API. Большинство разделов, на которые ссылается `make_docs/source/index.rst` (`basics/`, `advanced/`, `examples/`, `benchmarks/`, `api/`, `contribution`, `about`, `faq/`), физически не созданы.

?????? ???????????/???????????? ????

- **`mlumina/__main__.py`** — пустой стаб, `python -m mlumina` ничего не делает; в `pyproject.toml` нет `[project.scripts]`, т.е. CLI-команды у пакета нет.
- **`run.sh`** ссылается на `examples/isppr/examples.py` — такой директории (`examples/isppr/`) в репозитории нет.
- **`mlumina.core`** имеет побочные эффекты при импорте: `init_config(MLuminaConfig)` и `setup_logging()` вызываются на уровне `__init__.py`. Это требует, чтобы `configs/settings.yaml` (или файл по пути из `MLUMINA_SETTINGS_PATH`) существовал и был валиден, а `data.train_file`/`data.eval_file` из него — реально существующими файлами (тип `pydantic.FilePath`), иначе любой `import mlumina.core` (и транзитивно многое другое) упадёт при инициализации конфига.
- **`mlumina.scheduler.main`** тоже имеет побочный эффект при импорте: создаёт `Scheduler()` (подключение к Ray) и вызывает блокирующий `scheduler.start()` (запуск `uvicorn`). Модуль нельзя просто импортировать — только запускать как процесс.

- **`YoloImageModel.predict_ray`** / **`train_ray`** / **`tune_optuna`** внутри Ray-воркеров всегда создают `YOLO("yolov8n.pt")` как базовую архитектуру перед загрузкой переданных весов — корректно работает только для yolov8n-совместимых моделей, независимо от того, какая модель была реально обёрнута.
- **`xgboost_load_dataset`** — параметры по умолчанию (`target="elapsed_time_sec"`, `categorical_cols=["vcd", "nom_point"]`) специфичны для датасета из `examples/xgboost/`, а не универсальны.
- **`examples/`** в основном не проверяются автоматически: в `run.sh` рабочей (раскомментированной) остаётся только строка с `example_objects_simulate_ray.py`; остальные примеры нужно перед использованием проверять вручную.

Revision #1
Created 2026-09-23 06:40:15 UTC by Маслов Алексей
Updated 2026-09-23 06:40:16 UTC by Маслов Алексей