

Ниже — список **реально вычислимых** (на вашем симуляторе) предсказаний/объяснений, которые можно опереть на общедоступные данные. Я буду исходить из того, что у вас уже есть “MVP мира”:

- 1D-подложка + правила переписывания (SM-ядро),
- power-law граф геометрии ($\alpha \approx 2$) с лаплассианом $L\phi = \rho$,
- IFACE-embedding, OBS/IFACE, OBSFitness,
- режим FTL через wormhole-слой $H(t)$ как path-shortening,
- модель смысловой/наблюдательской динамики (t_{OT} , β_{sem} , $Q(t)$, $\lambda_s(t)$).

Ключевое: полезные предсказания должны быть **операциональными**: “что именно измерить” и “какую форму кривой/распределения ожидать”, а не философские.

0) Принцип предсказаний в вашей теории

У вас два уровня:

1. **Контекст=0 (совместимость):** мир должен воспроизводить стандартные наблюдения (GR/SM/QM) в пределах экспериментальных ошибок.
2. **Контекст=1 (расширенный режим):** появляются новые явления (wormhole/path-shortening, антиграв режимы, probability control). Это зона предсказаний: когда и как “аномалии” должны проявляться, и почему в обычных условиях мы их не видим.

Поэтому хорошая стратегия: найти предсказания, которые:

- в режиме контекст=0 дают **малые**, но специфические поправки;
- в режиме контекст=1 дают **редкие, но резкие** события (tail events).

1) Астрофизика/космология: что можно предсказать и проверить на открытых данных

1.1. Отклонения от ньютоновского $1/r^2$ на очень больших масштабах (не как MOND “в целом”, а как форма остатка)

У вас гравитация возникает из:

- дискретного power-law графа + лаплассиана
- и имеет “планковский” масштаб в hops (в модельных единицах).

Это почти неизбежно даёт **масштабные отклонения**:

- на ультрамалых масштабах (ниже ℓ_P в hops),
- и на ультрабольших (finite size / graph-renormalization).

Предсказание формы:

не “просто MOND”, а конкретный вид остатка в эффективном потенциале:

$$F(r) = GMr^2(1 + \delta(r)),$$

где $\delta(r)$ будет:

- почти 0 на промежуточном диапазоне,
- систематически отличаться от 0 за пределами «окна скейлинга»,
- причём форма $\delta(r)$ будет зависеть от параметров графа (α , структуры иерархий).

Как тестировать по открытым данным:

- кривые вращения галактик (SPARC),
- сильное/слабое гравитационное линзирование (DES/LSST/Planck),
- профили масс скоплений (CLASH).

Что именно сравнивать:

- не только “скорость vs r”, а форму отклонения $\delta(r)$ и её универсальность:
 - у вас $\delta(r)$ должна быть связана с масштабной структурой графа, поэтому предсказывается:
 - **одна и та же функциональная форма** $\delta(r)$ при масштабировании, а не произвольная галактика-по-галактике.

Вычислительная модель:

- построить набор “галактик” как распределения источников ρ на графике,
- вычислять ϕ решением $L\phi = \rho$,
- извлекать $F(r)$ и $\delta(r)$,
- обучить отображение параметров графа $\rightarrow \delta(r)$ и затем подгонять SPARC.

1.2. “Скрытая” эффективная размерность и её следствия: $D_{eff} \approx 4$ как объяснение некоторых космологических феноменов

В вашей валидированной геометрии:

- спектральная/фрактальная размерность графа может быть ≈ 4 , при этом наблюдатель видит 3D закон $F \sim 1/r^2$.

Предсказание:

существуют наблюдаемые эффекты “скрытой размерности” в **распространении и диссипации** полей/корреляций, не сводимые к 3D гладкому пространству.

Это может проявиться как:

- необычные скейлинги корреляционных функций,
- анизотропные или scale-dependent эффекты диффузии.

Данные:

- CMB (Planck): скейлинги корреляций на больших углах,
- large-scale structure (BOSS, DESI): power spectrum и bispectrum.

Модель:

- рассматривать распространение возмущений на вашем графе (random walk / heat kernel),
- измерять спектральную размерность $d_s(\ell)$ как функцию масштаба,
- сравнивать с наблюдаемыми скейлингами.

1.3. Гравитационные волны: дисперсия / задержка как функция “графовой геометрии”

Если геометрия — граф и embedding, то волны (возмущения ϕ или связанные поля) могут иметь:

- слегка модифицированную дисперсию,
- “скрытые пути” (wormhole режимы) в редких условиях.

Предсказание (контекст=0):

- очень маленькое, но специфическое отклонение скорости/дисперсии гравволн от с на больших дистанциях.

Данные:

- LIGO/Virgo/KAGRA: GWTC каталоги,
- сопоставление GW и EM counterpart (GW170817).

Это сложнее, но потенциально очень “публикуемо”, если найдёте сигнатуру $\delta t(f)$ в рамках модели.

2) Квантовая механика: предсказания и тесты

2.1. “Управление вероятностями” как экспериментальный предсказуемый эффект (если он реален)

Ваш FTL/Stone-механизм и Probability-control показывают, что:

- в детерминированном мире можно резко увеличить $P(\text{hit})$ через правильный action/context (M, Π_{meas}).

Предсказание:

существуют экспериментальные протоколы, где условное распределение исходов изменяется сильнее, чем допускает стандартная QM при фиксированном измерении — но только если measurement context действительно меняется (в терминах вашей модели).

Чтобы не конфликтовать с известными “no-signaling” тестами, важно, что:

- эффект должен быть:
 - контекстным,

- и проявляться как изменение ансамбля (post-selection / coarse graining / different POVM), а не как прямой “сдвиг вероятности при неизменном измерении”.

Практическая программа:

- выбрать класс квантовых экспериментов, где контекст измерения реально меняется:
 - weak measurement + post-selection,
 - delayed choice,
 - contextuality tests (KCBS, Peres–Mermin).
- построить модель Π_{meas} как в вашей симуляции:
 - разные coarse-projectors приводят к разным классам эквивалентности → разным статистикам.
- предсказать “узкие” места, где стандартная обработка данных скрывает эффекты, а NOBS-обработка (архивная) выявляет.

Данные:

- открытые datasets по квантовой томографии и contextuality есть, но часто нужно самим генерировать на установке (это не астрономия).

2.2. Предсказание о “семантической” границе наблюдателя: где QM переходит в классическое

У вас есть измеримые величины:

- $Q(t)$ (смысловая плотность),
- $\lambda_s(t)$ (скорость согласования),
- $\beta_1_{\text{sem}}(t)$ (топология понимания),
- а также phase saturation в RSL-логике.

Предсказание:

граница “классическое/квантовое” — не только физическая (масса/температура), но и информационная:

- когда наблюдатель не способен удерживать когерентное разбиение Π_{meas} (или фазовую структуру), интерференция исчезает.

Это можно попробовать сопоставить с:

- макроскопическими тестами интерференции (C60, большие молекулы),
- квантовая когерентность в биосистемах (спорно, но есть данные),
- рост decoherence rate vs complexity.

3) Что реально можно сделать “завтра”: 3 вычислительных проекта-предсказания

Ниже три направления, где вы можете быстро получить осмысленный результат.

Проект А: “Графовая гравитация vs SPARC (галактические кривые вращения)”

Цель: объяснить кривые вращения без тёмной материи как следствие $\delta(r)$ из графовой геометрии.

Пайплайн:

1. Взять SPARC (публично).
2. Для каждой галактики:
 - задать ρ (распределение масс) → источники на графике.
3. Решить $L\phi=\rho$ на вашем графике (или на семействе графов).
4. Предсказать $v(r)$ и сравнить с данными.
5. Фитить параметры графа (не RULESET!) на уровне нескольких гиперпараметров: α , степень иерархии, нормировка.

Выход: “универсальная кривая поправки” $\delta(r)$ и вероятность публикации — высокая, если получится устойчиво.

Проект В: “Спектральная размерность мира и космологические корреляции”

Цель: связать $d_s(\ell)$ вашей графовой геометрии с наблюдаемыми скейлингами LSS/CMB.

Пайплайн:

1. На графике вычислять heat kernel / return probability $P(t)$.
2. Оценивать d_s из:
$$P(t) \sim t^{-ds/2}$$
3. Смотреть, как d_s меняется по масштабу (многомасштабность).
4. Сравнивать с power spectrum наклонами и переходами режимов.

Проект С: “Probability control в квантовом протоколе (симуляция + лаборатория)”

Цель: найти протокол, где ваш Stone-подход даёт $P_F/P_0 \gg 1$ при смене контекста измерения, но не нарушает no-signaling.

Пайплайн:

1. Сначала сделать квантовую симуляцию (QuTiP) протокола с двумя POVM.
2. Определить TargetSpec (например, попадание в редкое подпространство при пост-селекте).
3. Реализовать `find_action_star` над space действий = {выбор POVM, выбор пост-селекции}.
4. Предсказать увеличение вероятности (это будет “контекстное управление вероятностью”).
5. Если получится, это можно повторить на реальной установке (фотоны).

4) Что добавить в симулятор для “реальных” предсказаний

Чтобы делать полезные предсказания, в симуляторе важно иметь:

1. Библиотеку forward-моделей наблюдений:

- как из ϕ и ρ получить $v(r)$, lensing $\kappa(r)$, GW задержки.

2. Калибровку единиц:

- сейчас всё в безразмерных hops; нужно map to physical units через:
 - подбор одного масштаба (например, $\ell_P \leftrightarrow$ Planck length, или $k=512 \leftrightarrow$ определённая физическая длина),
 - или через matching на одном наблюдаемом феномене.

3. Статистику устойчивости:

- predictions должны быть robust по начальным условиям и по окнам fit.

4. Версионирование мира:

- мир = (RULESET, graph generator, observer Π), и вы должны фиксировать hashes для воспроизводимости.

5) Какие открытые проблемы физики ваша теория потенциально объясняет

По текущему состоянию модели наиболее перспективные:

- “Почему $1/r^2$?” — уже конструктивно объясняется через графовую геометрию.
- “Почему 3D пространство?” — IFACE как embedding графа.
- “Тёмная материя/аномалии вращения” — $\delta(r)$ как следствие графовой геометрии/finite size.
- “Границы квантовой когерентности” — как информационный эффект $\Pi_{meas}/\text{phase saturation}$.
- “FTL/антиграв” — как контекстные режимы геометрии (wormholes/доп. поле χ), но это уже новая физика и должно быть тщательно отделено от совместимого режима.

Следующий шаг

Если выберете один из проектов А/В/С, я могу:

- сформулировать точный TargetSpec и метрики,
- предложить конкретный набор симуляционных параметров и выходных величин,
- набросать структуру репозитория и скриптов (data ingestion → simulation → fit → report),
- и указать, какие численные тесты нужно пройти, чтобы результат выглядел научно серьёзным (и воспроизводимым).

Сообщение 1/3 — Эксперимент (Проект) А из Experiments_v1.pdf

“Графовая гравитация vs SPARC (галактические кривые вращения)”

Ниже — максимально подробное описание именно того, что в файле названо **Проект А**: “объяснить кривые вращения без тёмной материи как следствие $\delta(r)$ из графовой геометрии”.

1) Научная идея и проверяемая гипотеза

1.1. Исходная гипотеза модели

В Experiments_v1.pdf сформулировано, что:

- гравитация у вас возникает из **дискретной графовой геометрии** (power-law граф, embedding),
- потенциал/поле ϕ определяется **графовым лаплассианом** через уравнение вида: $L\phi=\rho$,
где ρ — источники (масса/плотность), L — лаплассиан графа.

Отсюда следует, что эффективная сила/ускорение отклоняется от ньютоновского закона не “как MOND вообще”, а через **остаточную поправку** $\delta(r)$ в форме:

$$F(r)=GMr^2(1+\delta(r)),$$

и ключевой объект предсказания — **форма $\delta(r)$** как следствие:

- конечного размера (finite size),
- и/или многоуровневости/иерархии графа,
- и/или параметров генератора графа (в тексте явным образом упомянуты гиперпараметры типа α , “степень иерархии”, “нормировка”).

1.2. Что именно считается успехом (фальсифицируемо)

Проект А успешен, если получится одновременно:

1. **Воспроизвести кривые вращения SPARC** (скорости $v(r)$ как функция радиуса) без добавления гало тёмной материи.
2. Получить **устойчивую/универсальную** поправку $\delta(r)$:
 - не индивидуально под каждую галактику “с десятком параметров”,
 - а при подгонке **нескольких** гиперпараметров графа (в тексте: “не RULESET!, а несколько гиперпараметров: α , степень иерархии, нормировка”).
3. Показать, что $\delta(r)$ не является артефактом конкретного метода оптимизации:
 - при разумных вариациях деталей дискретизации, embedding и численного решателя результат сохраняется.

Если же для каждой галактики нужны отдельные произвольные настройки графа/масштабов, либо $\delta(r)$ получается нестабильной и “пляшет” при смене численных деталей — это сильное опровержение заявленной “геометрической” природы эффекта.

2) Данные и наблюдаемые величины (что сравниваем с реальностью)

2.1. Набор данных SPARC

В [Experiments_v1.pdf](#) явно сказано: “Взять SPARC (публично)”.

SPARC даёт для каждой галактики:

- наблюдаемую кривую вращения $v_{obs}(r)$ с ошибками,
- радиальные профили вклада барионов (газ, диск, балдж), либо напрямую компоненты $v_{bar}(r)$ / фотометрические профили и M/L (зависит от конкретной поставки SPARC).

Важный момент: в вашей постановке “источник” ρ должен строиться из барионного распределения (масса) и маппиться на график.

2.2. Основной наблюдаемый выход модели

Модель выдаёт предсказание $v_{model}(r)$ и/или эквивалентно предсказывает профиль ускорения $a(r)$, из которого:

$$v(r) = r a(r).$$

Параллельно модель должна выдавать:

- $\delta(r)$ — “универсальную кривую поправки” (в тексте это названо ключевым выходом).

3) Формализация: как из графа получить $v(r)$

В терминах [Experiments_v1.pdf](#) пайплайн такой:

Шаг A1 — Построить источники на графике

“задать ρ (распределение масс) → источники на графике”.

Технически это означает:

- выбрать график $G=(V,E)$ с некоторой параметризацией (ниже),
- задать отображение “физический радиус r ” → “подмножество вершин/слоёв графа”,
- разложить барионную массу по вершинам: $\rho: V \rightarrow R \geq 0$.

Здесь возникает критический дизайн-выбор, который должен быть зафиксирован в эксперименте, иначе это будет “скрытый фит”:

- как именно вы сопоставляете радиальные кольца галактики узлам графа (биннинг, сглаживание),
- как учитываете толщину диска, газ и т.д.

Шаг A2 — Решить уравнение на графике

“Решить $L\phi=\rho$ на вашем графике (или на семействе графов)”.

Это центральный вычислительный блок:

- строится лаплассиан L (комбинаторный или нормированный — надо выбрать и зафиксировать),
- решается линейная система (с граничными условиями или фиксацией gauge, т.к. у лаплассиана есть нулевое собственное значение).

Результат: $\phi(v)$ на вершинах.

Шаг А3 — Извлечь эффективную силу/ускорение и $v(r)$

В непрерывной физике $a(r) \sim -\nabla\phi$. На графике нужен дискретный аналог:

- либо градиент по embedding (если embedding определён; в тексте упоминается IFACE/embedding как принцип, но для Проекта А это не написано — важно аккуратно выбрать),
- либо радиальная производная по “слоям/хопам”,
- либо энергия на ребрах/потоки (в зависимости от того, как вы интерпретируете ϕ).

Далее строите:

- $a(r)$,
- $v(r) = r a(r)$.

Шаг А4 — Сравнение с SPARC

“Предсказать $v(r)$ и сравнить с данными.”

Сравнение должно учитывать:

- ошибки SPARC по $v_{obs}(r_i)$,
- ковариации (если используете),
- и, если вы позволяете варьировать M/L , то это отдельный “астрофизический nuisance”, который нужно либо фиксировать (литературные значения), либо фитить честно и одинаково для всех моделей.

Шаг А5 — Фит параметров графа (гиперпараметры)

“Фитить параметры графа (не RULESET!) на уровне нескольких гиперпараметров: α , степень иерархии, нормировка.”

Смысл: вы **не** подгоняете законы мира (ruleset), а подгоняете **геометрию** как класс:

- α — параметр степенного закона (power-law) графа,
- “степень иерархии” — насколько многоуровневая структура/ренормализация,
- “нормировка” — перевод модельных единиц/масштаб потенциала в физические.

4) Экспериментальный дизайн: чтобы это было “научно серьёзно”

В тексте [Experiments v1.pdf](#) прямо заявлено, что шанс публикации высокий “если получится устойчиво”. Устойчивость здесь обеспечивается дизайном:

4.1. Разделение параметров на классы

1. **Глобальные параметры модели** (общие для всех галактик):

- α ,
- уровень иерархичности,
- правила построения графа/лаплассиана,
- глобальная нормировка (или 1–2 параметра калибровки единиц).

2. **Галактические параметры-входы (из данных):**

- барионный профиль массы/света/газа (из SPARC),
- расстояние, наклон (inclination) — если SPARC даёт.

3. **Нежелательные/астрономические nuisance-параметры**, которые часто присутствуют даже в стандартных фитах:

- M/L для диска/балджа (если не фиксируете),
- возможные малые поправки на систематики.

Правило “научной серьёзности”: не превращать nuisance в “скрытые регуляторы” $\delta(r)$.

4.2. Схема валидации (обязательна)

- **Train/validation split по галактикам**: подбираете глобальные параметры по части галактик, проверяете на остальных.
- **Leave-one-out** или k-fold: проверяет, что “универсальная $\delta(r)$ ” реально универсальна.
- Отдельно: тест “low-surface brightness” vs “high-surface brightness” (SPARC это позволяет): MOND-подобные объяснения часто ломаются на крайних классах — ваш подход должен показать предсказуемую деградацию или устойчивость.

4.3. Бейзлайны для сравнения

Чтобы результат выглядел не как “ещё одна подгонка”:

- Ньютоновская барионная модель без тёмной материи (как нулевая гипотеза).
- Стандартный Λ CDM-фит (например NFW halo) — как “практический потолок”.
- MOND/RAR-бейслайн (по возможности) — как “класс альтернатив”.

В [Experiments v1.pdf](#) это не перечислено, но без этого статья/препринт обычно не проходит рецензию.

5) Какие именно метрики и графики должны получиться на выходе

Текст говорит: “Выход: универсальная кривая поправки $\delta(r)$ ”.

Значит в отчёте/репозитории должны появиться:

1. **Для каждой галактики:**

- график $v_{obs}(r)$ vs $v_{model}(r)$,
- остатки $v_{obs}-v_{model}$ и χ^2 -вклад по радиальным точкам.

2. Популяционная метрика:

- распределение χ^2/dof по галактикам,
- сравнение с NFW и MOND (если добавите).

3. Универсальность $\delta(r)$:

- построение $\delta(r)$ для каждой галактики в одинаковых координатах (например, по r/R_d или по ускорению),
- проверка, что кривые укладываются в “узкий пучок” при одном наборе глобальных параметров.

4. Устойчивость $\delta(r)$:

- при смене численного решателя (CG/Cholesky),
- при изменении детализации графа,
- при изменении схемы дискретизации ρ .

6) Риски/ловушки, которые нужно прямо закрыть в постановке

6.1. “Граф как универсальный интерполятор”

Если граф слишком гибкий (много степеней свободы), он сможет подогнать что угодно. Поэтому:

- параметры графа должны быть низкоразмерными и глобальными,
- запрещены “галактика-специфические” настройки геометрии (кроме входных данных ρ).

6.2. Неоднозначность “радиуса” и embedding

Переход “узлы графа \leftrightarrow физическое расстояние” — потенциальный источник произвольности. Если embedding не фиксирован, можно получить любую эффективную метрику.

В рамках Проекта А нужно заранее заявить:

- какой именно “distance” используется: hops, embedding-distance, спектральная дистанция и т.п.,
- и почему именно этот выбор соответствует “наблюдаемому r ” в SPARC.

6.3. Граничные условия в $L\phi=\rho$

Лаплассиан требует:

- фиксации нулевой моды (например $\sum \phi = 0$),
- или выбора условного “потенциал на бесконечности 0” (на конечном графе это аналог).

Разные choices могут давать разные “дальние хвосты” $\delta(r)$. Это должно быть стандартизировано.

7) Минимальный MVP (что реально сделать “завтра”, по духу документа)

С учётом текста [Experiments_v1.pdf](#), MVP-версия Проекта А:

1. Берёте небольшой поднабор SPARC (например 10–20 галактик разных типов).
2. Фиксируете одну схему построения ρ на графе.
3. Генерируете семейство графов с параметрами (α , иерархия, размер).
4. Решаете $L\phi = \rho$ и получаете $v(r)$.
5. Грубый фит 2–3 гиперпараметров на train-subset, проверка на holdout.
6. Смотрите: появляется ли общая форма $\delta(r)$ и падают ли остатки относительно ньютоновской барионной модели.

Если уже на MVP видно, что модель не может устойчиво улучшить fit — дальше масштабировать не имеет смысла.

Если ок, следующим сообщением опишу Эксперимент В из [Experiments_v1.pdf](#) (“Спектральная размерность мира и космологические корреляции”) в таком же уровне детализации: гипотеза → измеряемые величины → точный расчёт $ds(\ell)$ через $P(t)$ → сопоставление с LSS/CMB → метрики, бейзлайны, риски.

Сообщение 2/3 — Эксперимент (Проект) В из [Experiments_v1.pdf](#)

“Границы квантовой когерентности как информационный эффект Пmeas / phase saturation”

В [Experiments_v1.pdf](#) Проект В обозначен как:

«Границы квантовой когерентности — как информационный эффект Пmeas/phase saturation.»

Ниже — операциональное (вычислимое в вашем симуляторе) описание: что симулировать, что измерять, какую форму кривых ожидать, как сравнивать с общедоступными данными/известными трендами.

1) Научная идея и проверяемая гипотеза

1.1. Интуиция проекта В

Вместо “фундаментального коллапса” или просто “декогеренции от среды”, ваша рамка предполагает, что ограничения когерентности появляются как **информационный эффект наблюдения/измерения**:

- есть некоторый измерительный/наблюдательский функционал Пmeas (в терминах вашей системы: OBS/IFACE, ObservationTime tOT, смысловая динамика/контекст, и т.п.);
- и есть режим **phase saturation**: фаза (или эффективная фаза) перестаёт “нести различимую информацию” при превышении порога сложности/масштаба, из-за чего интерференционные члены подавляются.

Проект В проверяет не “почему квантовая механика работает”, а **где именно и по какому закону** она перестаёт давать интерференцию в вашем SM-ядре/IFACE-слое.

1.2. Фальсифицируемая гипотеза

Существует **универсальная** (или почти универсальная) зависимость “меры когерентности” от:

- *размера/сложности системы* (число степеней свободы N, глубина переписываний, “информационный объём” состояния),
- *интенсивности/частоты измерения* (сила Pmeas, частота опроса, “насколько сильно OBS вмешивается”),
- *контекста* (совместимый режим vs расширенный; но для В лучше держаться контекст=0, иначе будет неотделимо от “новой физики”).

И эта зависимость имеет характерную форму **порогового подавления** (phase saturation), а не просто экспоненциального спада “от времени” как в тривиальной модели шума.

2) Что именно измеряем (операциональные величины)

Чтобы В был реально проверяемым, вам нужен один “скаляр когерентности”, который можно считать:

2.1. Основная метрика: видимость интерференции V

Для любого интерференционного протокола (двухщелевой аналог, Мах–Цендер, Ramsey/Spin-echo-подобный) стандартная метрика:

$$V = I_{\max} - I_{\min} / I_{\max} + I_{\min}$$

В симуляции:

- I — частота попаданий/вероятность выхода в детектор (или в один из выходных портов).

Ключ: в вашей интерпретации V будет функцией не только “времени” и “шума”, а параметров Pmeas и “информационной сложности” состояния.

2.2. Альтернативная/дополнительная метрика: подавление вне диагонали $|\rho_{ij}|$

Если в симуляторе есть матрица плотности или её аналог (даже эффективный), то можно мерить:

- $\sum_{i \neq j} |\rho_{ij}|$ или $\text{Tr}(\rho^2)$ (purity).

2.3. Параметры, которые должны управлять экспериментом

В рамках ваших “MVP мира” (как вы перечислили в сообщении) естественно варьировать:

- N: размер “квантовой системы” (число узлов/ячеек/локальных степеней свободы в SM-ядре);
- T: число шагов переписывания (глубина эволюции);

- fmeas: частота измерения (каждые k шагов применяем Пmeas);
- η : “сила” измерения (жёсткость проекции/сжатия; насколько сильно Пmeas изменяет состояние);
- “температура/шум” (если есть) — но лучше начать с нуля, чтобы отделить чисто информационный эффект.

3) Протокол симуляции (как поставить В в коде)

Проект В должен быть устроен как минимальный цифровой аналог интерференционного теста, где “без измерения” должна быть высокая интерференция, а “с измерением” — подавление.

3.1. Подготовка: базовый интерференционный сценарий

Выберите один из двух вариантов (оба годятся, важно зафиксировать один):

Вариант 1: “двуихпутевый” протокол на графике

1. Подготовить состояние, которое распадается на 2 “пути” (две макроскопически различимые ветви в терминах конфигураций/подграфов).
2. Дать ветвям эволюционировать раздельно T шагов.
3. Свести ветви (recombine) и измерить распределение исходов в детекторах.

Вариант 2: “кубит + окружение”

1. Есть степень свободы “путь/спин” (2 состояния).
2. Есть окружение размера N, которое может запутываться.
3. Вы делаете Ramsey-последовательность и меряете контраст осцилляций по фазе.

Смысл обоих вариантов — получить измеримую V.

3.2. Включение Пmeas: модель “измерения как информационной операции”

В терминах [Experiments_v1.pdf](#) и вашей архитектуры OBS/IFACE:

- Пmeas применяется периодически или непрерывно и **снижает доступную фазовую информацию** (phase saturation), что должно проявиться как падение V.

Практически в симуляторе это может быть реализовано как:

- coarse-graining состояния (сжатие описания),
- стохастическая проекция на подалгебру наблюдаемых,
- ограничение точности фаз/амплитуд (квантование фазы),
- “reset” части степеней свободы по правилам OBSFitness/IFACE (если это ближе к вашей реализации).

Важно: в TargetSpec нужно формально зафиксировать оператор Пmeas (пусть даже в коде), иначе эксперимент неповторяем.

3.3. План сканов параметров (обязателен)

Минимальный план:

- $N \in \{8, 16, 32, 64, 128, \dots\}$
- f_{meas} (или k) — например $k \in \{1, 2, 5, 10, 20, \infty\}$ (∞ = без измерения)
- $\eta \in \{0, 0.1, 0.2, \dots, 1\}$
- при фиксированном T , и отдельный прогон по T , чтобы отделить “накопление” от “порога”.

4) Ожидаемый вид результатов (формы кривых)

Проект В обещает не просто “падает когерентность”, а характерный режим насыщения/порога.

4.1. Основное предсказание: пороговое подавление $V(N)$

Ожидаем, что при увеличении N (сложность/инфоёмкость) видимость интерференции:

- близка к 1 на малых N ,
- затем **резко** переходит к малым значениям после некоторого N^* ,
- причём N^* сдвигается при изменении f_{meas} и η .

Типовая “сигнатура phase saturation”:

- не чистая экспонента $V \sim e^{-cN}$,
- а скорее “плато → излом → спад” (логистический/erf-переход, либо степенной хвост после порога).

Операционально вы это проверяете фитами:

- экспонента vs логистическая/пороговая модель,
- сравнение AIC/BIC.

4.2. Предсказание “частота измерения как контрольная ручка”

Если эффект именно информационный, то увеличение частоты/силы измерения должно:

- уменьшать N^* (порог наступает раньше),
- увеличивать крутизну перехода,
- либо вводить зависимость $V \approx V(N \cdot f_{meas})$ (collapse переменных).

4.3. Предсказание “аналог квантового Зенона/анти-Зенона”

Поскольку P_{meas} — активный элемент, должны проявиться режимы:

- при очень частом измерении система “застывает” в базисе наблюдения (Zenon-подобное подавление переходов),
- при промежуточной частоте — возможен анти-Зенон (ускорение распада когерентности).

Это даёт характерную **немонотонность** некоторой метрики (например, скорости потери V как функции f_{meas}).

5) Связь с общедоступными данными (что реально “сверить”)

В отличие от SPARC (Проект A), для квантовой когерентности “единий публичный датасет” менее стандартен, но есть два пути, которые обычно проходят как “опора на общедоступное”:

5.1. Сверка с известными опубликованными scaling-законами

Можно брать из литературы (публичные графики/цифровки):

- зависимость времени когерентности от массы/размера в интерферометрии крупных молекул (C60 и дальше),
- эксперименты с сверхпроводниковыми кубитами (T_2 vs параметры считывания),
- оптомеханика (decoherence vs температура/давление/масса).

Важно: вы не обязаны воспроизвести абсолютные числа (в “мировых единицах”), но обязаны воспроизвести **форму зависимости** и ранжирование режимов.

5.2. Сверка с “типовыми формами” (без чисел)

Если вы пока не mapпите модельные единицы к физическим, то минимально честная проверка:

- совпадение качественных режимов (порог, зеноновская область, скейлинг по N_{fmeas}),
- проверка, что без P_{meas} интерференция сохраняется (контроль).

6) Бейзлайны (с чем сравнить внутри симулятора)

Чтобы доказать, что эффект специфичен для P_{meas} , нужны сравнения:

1. **Без измерения:** $\eta=0$ или $fmeas=\infty$ — должно быть максимальное V .
2. **Простой шум:** добавьте банальный деполяризующий/фазовый шум с той же “мощностью” и покажите, что форма кривой другая (обычно гладкая экспонента без порога).
3. **Альтернативный оператор измерения:** если заменить P_{meas} на другой coarse-graining, исчезает/меняется сигнатура phase saturation.

7) Метрики качества и критерии успеха

7.1. Первичные метрики

- $V(N, fmeas, \eta)$ — основная поверхность отклика.
- “порог” $N^{*(fmeas, \eta)}$ — точка, где V падает ниже, например, 0.5.

7.2. Вторичные метрики

- крутизна перехода (параметр логистического фита),
- показатель “коллапса переменных” (насколько хорошо кривые V совмещаются при рескейле осей),
- устойчивость по seed’ам (должно быть не случайным эффектом).

7.3. Успех (минимальный)

- Обнаружен устойчивый порог/насыщение фазы при сканировании N и параметров измерения.
- Порог контролируем fmeas и η предсказуемым образом.
- Бейзлайны (шум/без измерения) не воспроизводят ту же структуру.

8) Риски/ловушки постановки (что нужно “закрыть” заранее)

1. Подмена “измерения” шумом.

Если Pmeas реализован как случайная примесь шума, результат не будет специфическим.

2. Слишком свободная параметризация Pmeas.

Если у Pmeas много ручек, вы сможете “нарисовать” любой порог. Нужна минимальная форма и фиксированный протокол.

3. Непрозрачность отображения “фаза” \leftrightarrow “состояние SM-ядра”.

Нужно чётко сказать, что считается фазой (или аналогом), иначе “phase saturation” будет только метафорой.

9) Minimal TargetSpec (как бы я зафиксировал В в репозитории)

Если следовать “следующему шагу” из [Experiments_v1.pdf](#), то TargetSpec для В должен включать:

- **Task:** симулировать интерференционный протокол и измерить V.
- **Inputs:** N,T,fmeas, η ,seed.
- **Core:**
 - SM-ядро эволюции,
 - оператор Pmeas (строго определённый),
 - измерительный readout (какой observable и как получаем I_{max/min}).
- **Outputs:**
 - V,
 - (опционально) purity/вне-диагональные элементы,
 - оценка N^{*} и параметры фита.
- **Acceptance tests:**
 - при $\eta=0$: V \rightarrow 1 (в пределах статистики),
 - при сильном измерении: V \rightarrow 0,
 - воспроизводимость по seed’ам (дисперсия ограничена).

Если хотите, в следующем сообщении (3/3) опишу Эксперимент С из [Experiments_v1.pdf](#) — тот, что связан с FTL/антиграв как контекстным режимом геометрии (wormholes/поле χ) — и отдельно покажу, как “аккуратно отделить” совместимый режим от расширенного (как в файле и подчёркнуто).

Сообщение 3/3 — Эксперимент (Проект) С из [Experiments_v1.pdf](#)

“FTL/антиграв как контекстные режимы геометрии (wormholes / доп. поле χ) — строго отделённые от совместимого режима”

В [Experiments_v1.pdf](#) Проект С обозначен как:

«FTL/антиграв — как контекстные режимы геометрии (wormholes/доп. поле χ), но это уже новая физика и должно быть тщательно отделено от совместимого режима.»

Это важное замечание: **C — не “подкрутить” A/B**, а отдельный тест “расширенного контекста” (контекст=1 из вашей формулировки), который обязан иметь:

- чёткие триггеры включения,
- предсказуемые сигнатуры,
- и строгие проверки, что в контекст=0 ничего не ломается (no-signaling/каузальность на наблюдаемом уровне, отсутствие “антиграв” в обычных условиях и т.п.).

Ниже — описание в том же формате, что А и В.

1) Научная идея и проверяемая гипотеза

1.1. Что утверждает проект С

У вас есть “режим FTL через wormhole-слой $H(t)$ как path-shortening” (вы это перечислили как часть MVP), а в [Experiments_v1.pdf](#) это сформулировано как:

- **контекстные режимы геометрии:** при определённых условиях активируется “wormhole-слой” или дополнительное поле χ ,
- что приводит к **сокращению пути** (path-shortening) и/или эффективному изменению знака/структуре гравитационного ответа (“антиграв” режим).

То есть С — это “геометрический переключатель”: в обычном режиме метрика/граф ведут себя стандартно, а в расширенном — появляются редкие “короткие связи” или изменённая динамика.

1.2. Фальсифицируемая гипотеза (операционально)

Существует функция/критерий активации:

$\text{context}=1 \Leftrightarrow T(\text{state}, \text{OBS}, \chi, \lambda s, Q, tOT, \dots) > \theta$

после чего:

1. **На графике возникает measurable path-shortening**, т.е. сокращение эффективных расстояний между определёнными регионами (в терминах hop-distance / geodesic / effective metric).
2. Это даёт **наблюдаемые сигнатуры** в динамике сигналов/частиц на графике:
 - время достижения цели (arrival time) становится меньше, чем допускает baseline-метрика,

- распределение времен/длин путей получает характерный “левый хвост” (редкие сверхбыстрые события),
- либо меняется знак/профиль отклика на “гравитационный” источник (антиграв-аналог).

Если при корректно заданных триггерах Т эффект не появляется — гипотеза о wormhole/ χ -режиме в вашем симуляторе не подтверждается даже на toy-мире.

2) Что именно измеряем (операциональные величины)

Проект С должен опираться не на “FTL как лозунг”, а на конкретные измеряемые на графике величины.

2.1. Метрика 1: эффективная дистанция / длина кратчайшего пути

Определяем:

- $d_0(u,v)$ — baseline расстояние (обычный граф, контекст=0),
- $d_1(u,v,t)$ — расстояние при активном $H(t)$ /wormhole-слое.

Тогда эффект:

$$\Delta d(u,v,t) = d_0(u,v) - d_1(u,v,t) > 0$$

Важны не единичные пары, а статистика по множеству пар.

2.2. Метрика 2: время распространения сигнала (first passage time)

Запускаем “сигнал” (волна/частица/маркер) из источника и меряем:

- T_{arrive} до цели,
- распределение $P(T_{\text{arrive}})$ по множеству запусков/пар.

Сигнатура FTL-подобия в toy-мире: существенная масса вероятности на временах, недостижимых без wormhole-слоя, то есть появление отдельного режима (“ранние приходы”).

2.3. Метрика 3: “антиграв” как смена знака/профиля отклика

Если у вас есть гравитационный потенциал/поле ϕ и ускорение $a \sim -\nabla\phi$, то “антиграв” можно формализовать как:

- смена направления ускорения относительно источника,
- или изменение зависимости $a(r)$ на участке (например, локально a указывает “от массы”).

В кодовой базе это очень близко к проверкам, которые видны во фрагменте [world_search_final\(2\).pdf](#): там явно сравнивают ускорение с $-\nabla\phi$ и проверяют эквивалентность. В С аналогично можно ввести проверку “знака” и “монотонности” профиля.

3) Протокол симуляции (как поставить С в коде)

3.1. Базовый мир (контекст=0) — обязательный контроль

Перед любым “FTL” нужно зафиксировать baseline:

1. Сгенерировать геометрию (ваш power-law граф, embedding).
2. Определить стандартную динамику (распространение, гравитационный ответ).
3. Замерить:
 - $d_0(u,v)$,
 - $T_{\text{arrive}}(u,v)$,
 - профиль $a_0(r)$ при стандартных источниках.

Этот baseline используется как “сравнить до/после”.

3.2. Определение “wormhole-слоя” $H(t)$

В терминах [Experiments v1.pdf](#) wormhole-слой — это механизм **path-shortening**.

Операционально это может быть реализовано как:

- добавление временных ребер (shortcuts) между удаленными узлами,
- временное изменение весов ребер (снижение “стоимости” перехода),
- включение отдельного подграфа $H(t)$, который соединяет кластеры.

Важно зафиксировать:

- **как выбираются пары узлов для wormhole-ребер** (случайно? по χ ? по OBS/IFACE?),
- **какая интенсивность** (сколько shortcut-ребер на узел),
- **какая длительность жизни** shortcut (сколько шагов).

3.3. Триггер контекста (когда включается $H(t)$)

Поскольку в вашей общей рамке “контекст=1” связан с наблюдательской/семантической динамикой, естественный вариант (и он согласуется по духу с [Meaning v1.pdf](#), гл. 14 про наблюдательные метрики) — сделать триггер функцией от:

- $\lambda(t) / \lambda_s(t)$: режимы “ $\lambda \approx 1$ ” vs “ $\lambda \gg 1$ ”,
- $Q(t)$: смысловая плотность/нагрузка,
- tOT: моменты “коллапса инварианта” (стационарность метрики когерентности),
- и/или некоего поля χ .

Пример того, что должно быть явно прописано в TargetSpec:

- $H(t)$ включается только на окнах времени, когда выполнено $\lambda(t) \approx 1$ и система близка к ОТ (или наоборот — в зависимости от вашей модели);
- либо $H(t)$ включается при превышении порога $\chi > \chi^*$.

Без явного триггера С превращается в “мы включили shortcut когда захотели”.

4) Ожидаемый вид результатов (формы кривых/распределений)

4.1. Сигнатура path-shortening: распределение Δd

После активации $H(t)$ ожидаете:

- Δd распределено не как шум вокруг 0, а имеет:
 - либо явный сдвиг среднего $E[\Delta d] > 0$,
 - либо тяжёлый хвост больших Δd (редкие “мосты” дают огромный выигрыш).

График: CCDF $P(\Delta d > x)$ должен показывать дополнительный хвост.

4.2. Сигнатура FTL-подобия: смесь распределений времен прихода

Для T_{arrive} ожидается не просто ускорение всех путей, а **двухрежимность**:

- основной пик соответствует обычной геометрии,
- второй пик/хвост слева — редкие сверхбыстрые приходы.

То есть $P(T)$ лучше описывается смесью:

$$P(T) \approx (1-p)P_0(T) + pP_{\text{wh}}(T),$$

где p — частота активации/попадания в wormhole-маршрут.

4.3. Сигнатура “антиграв”: смена направления ускорения / немонотонность $a(r)$

Если реализуете “антиграв” как эффект поля χ , то у источника массы при контекст=1 может появляться область, где:

- $a(r)$ меняет знак,
- или da/dr нарушает обычную монотонность,
- или “пробная частица” уходит от источника.

Критически важно показать:

- что это не численная нестабильность решателя $L\phi = \rho$,
- и что в контекст=0 знак всегда “нормальный”.

5) “Тщательно отделить от совместимого режима”: какие нужны тесты безопасности

Фраза из [Experiments v1.pdf](#) — центральная: “это уже новая физика и должно быть тщательно отделено от совместимого режима.”

Это значит: в эксперименте С обязательно должны быть **гейты**, доказывающие, что эффект не “просачивается” в baseline.

5.1. Тест 1: нулевая утечка при контекст=0

При выключенном триггере:

- никаких shortcut-ребер,

- статистика $d(u,v)$, $Tarrive$, $a(r)$ совпадает с baseline (в пределах Monte-Carlo ошибок).

5.2. Тест 2: локальность условий активации

Если $H(t)$ зависит от OBS/семантики, нужно показать:

- что “дальний наблюдатель” (без доступа к контексту) не может использовать это для сигнальной передачи в контекст=0.

Практически: вы вводите протокол A/B:

- в А “наблюдатель пытается включить контекст=1”,
 - в В не пытается,
- и проверяете, что без доступа к “координатам контекста” отличить А от В нельзя (или отличие проявляется только после пост-селекции/локального доступа). Это концептуально близко к тому, как в квантовых задачах избегают прямого no-signaling нарушения.

5.3. Тест 3: воспроизводимость и редкость (tail events)

Если эффект редкий, вы обязаны дать:

- оценку частоты p ,
- доверительный интервал,
- и зависимость $p(\theta)$ от порога триггера.

Иначе любой “разовый” сверхбыстрый приход будет выглядеть как случайность.

6) Как привязать С к наблюдаемым/открытым данным (честно)

Проект С сложнее привязать к “готовым” публичным наборам (как SPARC) без лаборатории. Но есть реалистичные варианты, которые остаются в рамках “операционности”:

6.1. Симуляционный “публичный результат”: универсальные распределения

Даже без внешних данных вы можете опубликовать:

- распределения $P(Tarrive)$, $P(\Delta d)$,
- фазовые диаграммы $p(\eta, \theta)$ активации,
- и строгие контрольные тесты контекст=0.

Это будет “computational physics result”: проверяемость — через репозиторий и воспроизводимость (как вы и пишете в [Experiments v1.pdf](#): hashes, версии мира, тесты).

6.2. Связь с [Meaning v1.pdf](#): наблюдательская часть как триггер

Если вы хотите, чтобы “контекст” был не абстрактным тумблером, а связанным с наблюдателем, то можно использовать метрики из главы 14 [Meaning v1.pdf](#):

- $\lambda(t)$,
- внутренняя/межличностная когерентность $\Xi_{\text{внутр}}(t)$, $\Xi_{\text{меж}}(t)$,

- OT как точки стационарности $d\Xi/dt \approx 0$.

Тогда эксперимент C можно поставить как “гибридный”:

симулятор мира + “триггерные” временные ряды (реальные или синтетические), которые включают/выключают $H(t)$. Даже если пока это не физический FTL, это даёт чёткий протокол включения, который можно повторять.

7) Метрики качества и критерии успеха

7.1. Первичные метрики

- Δd статистика: среднее, медиана, хвост CCDF.
- T_{arrive} : доля “ранних приходов” r и параметры смеси распределений.
- Для антиграв: доля траекторий с “отталкиванием” от источника + профиль $a(r)$ до/после.

7.2. Успех (минимальный)

- Есть чётко определённый триггер контекста.
- При контекст=1 появляется статистически значимое path-shortening (не исчезает при смене seed’ов).
- При контекст=0 эффект отсутствует (жёсткий null test проходит).

7.3. Провал

- Эффект появляется “иногда” и также появляется в baseline.
- Результат чувствителен к деталям решателя/дискретизации (численная нестабильность).
- Нельзя воспроизвести распределения при фиксированных хэшах “мира”.

8) Minimal TargetSpec для C (в духе “следующего шага” из [Experiments_v1.pdf](#))

Task: обнаружить и количественно описать режим path-shortening/антиграв при контекст=1.

Inputs:

- параметры графа/геометрии (как в A),
- параметры wormhole-слоя $H(t)$: плотность shortcut, время жизни,
- параметры поля χ (если используется),
- правило триггера $T > \theta$,
- seed’ы.

Procedure:

1. baseline прогон контекст=0: собрать $d_0, T_{\text{arrive}0}, a_0(r)$.
2. прогон контекст=1: включать $H(t)$ по триггеру, собрать $d_1, T_{\text{arrive}1}, a_1(r)$.
3. статистическое сравнение + null tests.

Outputs:

- таблицы/файлы с $P(\Delta d)$, $P(T)$,
- оценка p и её доверительный интервал,
- графики профилей $a(r)$ и тесты на смену знака/аномалию.

Acceptance tests:

- “No-leak”: в контекст=0 $E[\Delta d] \approx 0$, нет ранних приходов.
- “Activation”: при контекст=1 $E[\Delta d] > 0$ и/или p существенно > 0 .
- “Robustness”: эффект сохраняется при изменении seed и умеренных изменениях численной реализации.

Если вы хотите, могу в следующем шаге сделать для С “конкретику уровня кода”:

- предложить точную формулу триггера T на основе $\lambda(t)$ и $\Xi(t)$ из [Meaning_v1.pdf](#),
- и расписать минимальный набор модулей репозитория ($world \rightarrow trigger \rightarrow wormhole \rightarrow metrics \rightarrow report$) так же, как вы планируете в [Experiments_v1.pdf](#).

Ниже — “конкретика уровня кода” для **Эксперимента C** (FTL/антиграв как контекстный wormhole-слой $H(t)$), с (1) точной формулой триггера T на основе $\lambda(t)$ и $\Xi(t)$ из [Meaning_v1.pdf](#) и (2) минимальной структурой репозитория по цепочке $world \rightarrow trigger \rightarrow wormhole \rightarrow metrics \rightarrow report$, как запрошено в [Experiments_v1.pdf](#) и с учётом двухрежимной валидации из [New_Physics_v1.pdf](#).

1) Точный триггер $T(t)$ из $\lambda(t)$ и $\Xi(t)$ ([Meaning_v1](#))

1.1. Что берём из [Meaning_v1.pdf](#)

В файле задаётся идея, что:

- $\lambda(t)$ — “скорость согласования / режим наблюдения” (есть режимы $\lambda \approx 1$ и $\lambda \gg 1$);
- $\Xi(t)$ — метрика когерентности (внутренняя/межличностная);
- “Observation Time / OT” — момент/окно, где метрика когерентности выходит на стационарность (условно $d\Xi/dt \approx 0$).

Для С нам нужно **детерминированное**, редкое, “ресурсно дорогое” условие включения $H(t)$, как требуется в [New_Physics_v1.pdf](#): $context=0 \rightarrow H(t)=\emptyset$, и только в device-режиме (контекст=1) триггер может сработать.

1.2. Нормализация величин

Чтобы формула была устойчива, вводим нормированные версии:

- $\lambda_n(t) = \log(1 + \lambda(t))$ — чтобы режим $\lambda \gg 1$ не доминировал численно.
- $\Xi(t) \in [0, 1]$ — если у вас не так, приводим линейно:

$$\Xi_n(t) = \text{clip}(\Xi(t) - \Xi_{\min}, \Xi_{\max} - \Xi_{\min}, 0, 1)$$
- Производная на окне (чтобы не реагировать на шум):

$$\dot{\Xi}W(t) = \Xi(t) - \Xi(t-W)W$$
- “Стационарность OT”:

$$SOT(t) = \exp(-|\dot{\Xi}W(t)|\epsilon\Xi) \in (0, 1]$$
 где $\epsilon\Xi$ — допуск “почти стационарно”.

1.3. Триггер T(t): однозначная формула

Требования (по смыслу С и [New_Physics_v1.pdf](#)):

- редкость,
- детерминированность,
- зависимость от наблюдательского режима,
- явное отличие baseline vs device.

Предлагаю следующий **скаляр триггера**:

$T(t) = SOT(t) \cdot \sigma(\Xi n(t) - \theta \Xi \tau \Xi) \cdot \text{достаточная когерентность} \cdot \sigma(\theta \lambda - \lambda n(t) \tau \lambda) \cdot 1[\text{device_mode}=1]$ гейт совместимости

где:

- $\sigma(x) = 1/(1 + e^{-x})$ (логистическая функция),
- $\theta \Xi$ — порог когерентности (например 0.7),
- $\theta \lambda$ — порог “не слишком большой λ ” (т.е. близость к $\lambda \approx 1$ в лог-шкале),
- $\tau \Xi, \tau \lambda$ — “мягкость” порога (например 0.05–0.15),
- SOT делает активацию возможной только около ОТ-окон (стационарность когерентности),
- индикатор `device_mode` гарантирует: **в baseline С никогда не активируется** (как и требует “Compatibility mode” из [New_Physics_v1.pdf](#)).

Решение включения:

`activate(t)=1[T(t)>θT]`

где θT обычно 0.8–0.95 (делает событие редким).

1.4. Почему именно так (коротко, но по делу)

- Ξ высокая: у наблюдателя/системы есть “согласованная структура”, которую можно использовать как “устройство”.
- $\lambda \approx 1$: режим “не разогнанной” семантической динамики (снижаем риск неконтролируемой “Both/N” логики).
- ОТ-стационарность: фиксирует “замыкание петли” (идемпотентность/стабилизация наблюдения по смыслу Meaning).
- `device_mode` — формальная реализация разделения совместимости/аномалии из [New_Physics_v1.pdf](#).

1.5. Минимальный код триггера (псевдо-Python)

```
import math

def sigmoid(x: float) -> float:
    return 1.0 / (1.0 + math.exp(-x))

def trigger_T(t, lambda_series, xi_series, w,
             eps_xi=1e-3,
             theta_xi=0.7, tau_xi=0.08,
             theta_lam=math.log(2.0), tau_lam=0.08, # log(1+lambda) ~ log(2) =>
lambda~1
             device_mode=False):
    if not device_mode:
        return 0.0
```

```

lam = lambda_series[t]
Xi = Xi_series[t]
Xi_prev = Xi_series[max(0, t-w)]

lam_n = math.log(1.0 + lam)
Xi_n = max(0.0, min(1.0, Xi)) # если уже [0,1]

dXi = (Xi - Xi_prev) / max(1, w)
S_ot = math.exp(-abs(dXi) / eps_xi)

gate_xi = sigmoid((Xi_n - theta_xi) / tau_xi)
gate_lam = sigmoid((theta_lam - lam_n) / tau_lam) # high when lam_n <=
theta_lam

return S_ot * gate_xi * gate_lam

def activate(t, **kwargs):
    T = trigger_T(t, **kwargs)
    return T > 0.9

```

2) Минимальный набор модулей репозитория: world → trigger → wormhole → metrics → report

Ниже — структура, которая прямо отражает “TargetSpec → симуляция → метрики → отчёт”, о чём говорится в [Experiments v1.pdf](#), и включает двухрежимную валидацию из [New Physics v1.pdf](#).

2.1. Дерево папок (минимальный “инженерный скелет”)

```

repo/
    README.md
    pyproject.toml (или requirements.txt)

world/
    __init__.py
    state.py          # WorldState: X_t, phi, P_t, S_sem, flags
    graph.py         # GraphStructure + базовые операции
    physics.py       # baseline динамика (RSL-совместимая)
    embedding.py     # IFACE embedding, distance, optional kd-tree
    rng.py           # seed/контексты ω (детерминированные)

control/
    __init__.py
    trigger.py       # T(t), activate(t) на основе lambda, Xi
    wormhole.py      # build_wormhole_edges / apply_H_t
    actions.py       # ActionGeometry, ActionMeasureContext (если нужно)
    device_mode.py   # gate: baseline vs device

metrics/
    __init__.py
    ftl.py           # d0/d1, Δd, arrival-time, mixture fit
    antigrav.py      # grad_phi_fn, sign checks, a(r) профили
    compatibility.py # Born rule corr, F~1/r^2, инварианты SM
    paradox.py       # paradox_penalty: β1_sem, циклы, нестабильность

experiments/
    __init__.py
    expC_protocol.py # два режима валидации A/B (как в New_Physics)
    targets.py        # TargetSpecFTLDelivery / TargetSpecAntiGrav
    sweep.py          # прогон по seeds, budgets, порогам

```

```

data/
    contexts/           # (опционально) сохранённые lambda(t), xi(t) или
    синтетика
    runs/              # артефакты прогонов (parquet/json)

reports/
    make_report.py     # агрегация метрик → figures/tables
    templates/          # md/tex шаблоны

tests/
    test_trigger.py
    test_wormhole_gating.py
    test_compatibility_mode.py
    test_reproducibility_hash.py

```

3) Что делает каждый слой (по цепочке)

3.1. world/ — “обычная физика” (контекст=0)

Цель: реализовать совместимый базис, который всегда воспроизводим и не содержит $H(t)$.

- world/state.py
 - WorldState хранит:
 - X_t (состояние SM-ядра/подложки),
 - ϕ (если есть),
 - S_{sem} (семантическое состояние, если участвует),
 - P_t (если вы добавили вероятность как объект состояния — это из [New Physics v1.pdf](#)),
 - mode / device_mode.
- world/physics.py
 - step_baseline(state) → state — детерминированный шаг RSL-мира без wormholes.

Инвариант-гейт: при `device_mode=False` любой код wormhole/FTL не должен влиять ни на граф, ни на метрики.

3.2. control/trigger.py — вычисление $T(t)$ и флага activate

Берёт:

- временные ряды $\lambda(t)$, $\Xi(t)$ (из [Meaning v1.pdf](#) либо синтетические),
- параметры порогов/окон,
- и выдаёт:
- T_{value} ,
- `activate=True/False`.

Важно: этот модуль **не должен** трогать физику; он только вычисляет “контекстный сигнал”.

3.3. control/wormhole.py — построение $H(t)$ (wormhole-edges) детерминированно

Требование из [New Physics v1.pdf](#):

$H(t)=FH(X(t),Ssem(t),context)$, и при $context=0 \rightarrow H(t)=\emptyset$.

Интерфейс:

- `build_wormhole_edges(state, budget, anchor_type, ...) -> list[Edge]`
- `apply_H_t(graph, edges) -> graph2` или “временное наложение” без мутации исходного.

Детерминированность: edges выбираются из state + фиксированных rules, без RNG (или RNG только через seed контекста ω , который является частью детерминированного эпизода).

Минимальная реализация выбора якорей:

- `anchor_source = argmax distance in IFACE`
- `anchor_target = argmax distance from source`
(или через “OmegaTypeX”, как упоминается в [New Physics v1.pdf](#), но это уже ваш словарь типов).

3.4. metrics/ftl.py — что считать “FTL” в симуляторе

В [New Physics v1.pdf](#) явно сказано: FTL надо определить на уровне IFACE как превышение скорости относительно локального канала; аккуратно: не “частица быстрее”, а “маршрут короче”.

Минимально:

- `d0 = iface_distance(graph_baseline, u, v)`
- `d1 = iface_distance(graph_with_H, u, v)`
- $\Delta d = d0 - d1$
- “FTL-событие” = $\Delta d \geq \Delta d_{min}$ (или отношение $d1/d0 \leq r_{max}$).

Параллельно arrival-time:

- симулировать распространение (random walk / wavefront / message passing) и измерять `T_arrive`.

3.5. metrics/paradox.py — paradox_penalty (логическая катастрофа)

В [New Physics v1.pdf](#) фитнес включает:

$F=wAFA+wBFB-wP$ paradox_penalty

и подчёркнуто, что логика четырёхзначная (T/F/B/N), и штрафовать надо **неконтролируемое** “Both” там, где нужна однозначность.

Значит `paradox_penalty` должен как минимум учитывать:

- рост “Both”/“Neither” в compatibility-проверках,
- невозможность стабилизации (например, отсутствие OT/стационарности),
- появление патологических циклов.

Даже если $\beta_{1,\text{sem}}$ пока не реализован, можно начать с прокси:

- доля проверок, переходящих в B/N в baseline-режиме,
- количество шагов без стабилизации $|\Xi'| > \epsilon$.

4) Двухрежимная валидация A/B (как в New_Physics) — “протокол эксперимента C”

4.1. Режим А: Compatibility mode (baseline)

Файл [New_Physics_v1.pdf](#) перечисляет тесты вида:

- Born rule corr ≈ 1 ,
- $F \sim 1/r^2$ в окне,
- SM-инварианты.

Для С это означает:

- `device_mode=False`
- $H(t)$ всегда пуст
- metrics:
 - `compatibility_score` высокий,
 - `ftl_score` должен быть нулевой (или строго ниже порога),
 - `paradox_penalty` ≈ 0 .

4.2. Режим В: Device / anomalous mode

- `device_mode=True`
- включается триггер $T(t)$ и, если `activate`, строится $H(t)$
- цель:
 - `ftl_score` высокий (ранние приходы / $\Delta d > 0$),
 - но `paradox_penalty` не взрывается,
 - и baseline-инварианты не ломаются “глобально” (например, вне окон активации всё возвращается к норме).

5) Минимальный “run script” (sweep) для получения $P\Phi/P0 \gg 1$

Из [New_Physics_v1.pdf](#) следует практический рецепт: контексты $\omega = \text{seeds}$, считаем вероятность успеха цели до/после action.

В терминах С:

- Определяем событие успеха Φ : “доставить сигнал из А в В быстрее baseline-предела”.
- Для каждого seed (контекст ω) запускаем эпизод:
 - baseline action: `ActionGeometry(mode="baseline")`
 - device action: `ActionGeometry(mode="wormhole", budget=b) + включённый trigger`

Считаем:

- $P_0 = \Pr[\Phi | \text{baseline}]$
- $P_\Phi = \Pr[\Phi | \text{device}]$
- ratio P_Φ/P_0

6) Минимальные тесты (чтобы это выглядело воспроизводимо и “научно строго”)

1. **Test gating:** при `device_mode=False` функция `build_wormhole_edges` возвращает пусто при любых входах.
2. **Test determinism:** при фиксированном `seed` и одинаковых `series` λ, Ξ набор wormhole-ребер идентичен.
3. **Test no-leak FTL:** в `baseline` распределение Δd имеет массу около 0 и не имеет “левого хвоста” времен прихода.
4. **Regression test на триггер:** заранее сохранённые короткие `series` $\lambda, \Xi \rightarrow$ ожидаемые `activate` на определённых `t`.

Если ответите на 3 вопроса из [New_Physics_v1.pdf](#) (они там прямо перечислены):

1. есть ли `i_center` в `IFACEObject`,
2. можно ли добавлять ребра на лету и как устроены `neighbors`,
3. где живёт `phi`,
— я адаптирую это под ваши реальные имена классов и дам уже не псевдо-, а “вставляемый” код: `build_wormhole_edges(...)`, `grad_phi_fn(...)`, `TargetSpecFTLDelivery(...)` и готовый `experiments/expC_protocol.py`.