

PICSFORDESIGN | MACHINE LEARNING ARCHITECTURE

PFD Hybrid AI Product Generator

Русский | English | 简体中文

Language	Section
Русский	Гибридная архитектура, workflow, LoRA, QA, cost control и roadmap
English	Hybrid architecture, workflow, LoRA, QA, cost controls and roadmap
简体中文	混合架构、工作流、LoRA、质量控制、成本保护与实施路线图

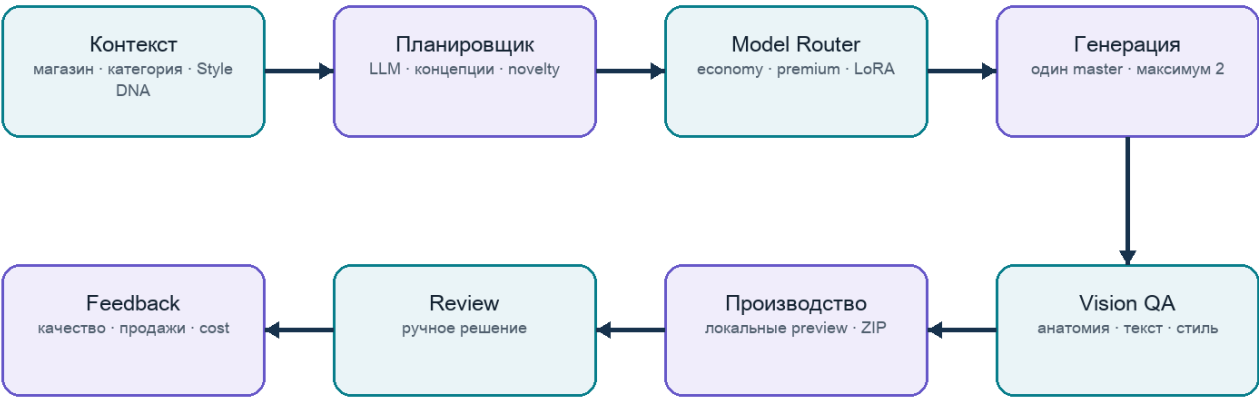
Version 1.0 | 5 October 2026 | Internal technical design for PFD AI Studio

PICSFORDESIGN | RU

Архитектура гибридного AI генератора продуктов PFD

Первый интегрированный контур обучения LoRA и автоматизации

2 Гибридная архитектура



Feedback питает память PFD, dataset и LoRA training; Model Registry возвращает утверждённую версию в Router.

Рисунок 1. Гибридный workflow PFD от контекста до Model Registry

Документ определяет, как PicsForDesign строит гибридный AI генератор продуктов. Готовые текстовые и графические модели выполняют базовую генерацию, а собственный контур PFD управляет контекстом, памятью, уникальностью, качеством, стоимостью и обучением LoRA адаптеров.

1 Главное архитектурное решение

PFD не обучает foundation model с нуля. Первая версия использует внешние модели через API и собственный оркстратор. После накопления лицензированного и проверенного набора данных PFD обучает компактные LoRA адаптеры для конкретных стилей, магазинов и постоянных персонажей. Собственной обученной моделью PFD считается связка версии LoRA адаптера, версии dataset, model profile и утверждённого набора метрик.

3 Уровни адаптации и обучения

Уровень	Механизм	Обучение весов	Назначение	GPU
A	Prompt rules и structured context	Нет	Формат, стиль, ограничения, tone of voice	Нет
B	RAG, embeddings, references, control adapters	Нет	Память, поиск примеров, композиция и identity	Не обязательно
C	LoRA fine tuning	Да, малый adapter	Закрепление стиля или постоянного персонажа	Да для обучения

4 Текущее и целевое состояние

Область	Сейчас	Цель
Генерация	Один master через gpt-image-1-mini low	Router между economy, premium и LoRA
Стоимость	Не более двух платных попыток	Budget ledger и остановка до API call
Превью	Локальные производные из master	Шаблоны по типу продукта и каналу
QA	Vision QA и visual uniqueness gate	Профили QA для людей, персонажей и creative kits
Память	Archive memory и metadata Style DNA	Vector retrieval, Character Library и feedback memory
Обучение	Не выполняется	Версионизируемый LoRA training и Model Registry

5 Полный workflow генерации продукта

Шаг	Этап	Алгоритм	Выход и gate
01	Триггер	Расписание, ручной запуск или API создают run с idempotency key	Задача создана один раз
02	Бюджет	Проверяются дневной лимит, лимит продукта и credits	Разрешено или blocked_budget
03	Права	Проверяются права на архив, characters и training assets	Разрешено или blocked_rights
04	Контекст	Магазин, категория, аудитория, язык, формат и канал	Context snapshot
05	Style DNA	Палитра, темы, словарь, композиции и запрещённые признаки	Versioned style brief
06	Character Profile	Identity references, одежда, возраст и допустимые вариации	Character brief или none
07	Память	Vector retrieval выбирает разнообразные работы и feedback events	Reference set
08	Концепции	LLM создаёт минимум три существенно разные идеи	Concept candidates
09	Novelty текста	Сравниваются title, description, tags и concept embeddings	Novelty score
10	Выбор	Ranker балансирует релевантность, новизну, риск и бюджет	Selected concept
11	Prompt	Создаются prompt, negative constraints, QA rubric и outputs	Prompt package
12	Router	Выбирается economy, premium или LoRA profile	Model profile
13	Master	Генерируется одно изображение n = 1	Master candidate
14	Safety	Проверяются формат, запрещённый контент, текст и watermark	Pass или reject
15	Vision QA	Анатомия, лицо, стиль, композиция и целостность элементов	QA report
16	Image novelty	Image embedding и perceptual hash сравниваются с архивом	Visual novelty score
17	Коррекция	Одна финальная targeted repair попытка	Second candidate или blocked
18	Derivatives	Thumbnail, gallery и marketplace размеры создаются локально	Пакет без нового API cost
19	Metadata	Title, description, tags, license и manifest	Draft package
20	Human review	Редактор принимает, отклоняет или правит с reason code	Decision event
21	Публикация	Одобрённый пакет получает product ID	Published product
22	Feedback	Решение, продажи, дефекты, правки и стоимость	Training evidence
23	Dataset	Только проверенные примеры с правами входят в новую версию	Candidate dataset
24	LoRA cycle	Training, offline eval, canary, promotion или rollback	Approved adapter

6 Model Router

Router выбирает не самую новую, а минимально дорогую модель, способную пройти заданный порог качества. Неопределённые и высокорисковые задачи переводятся в ручной режим.

Профиль	Когда	Модель	Ограничение
Draft Economy	Идеи и низкорисковые creative kits	Недорогая API image model	Один master, максимум 2 попытки
Premium Master	Hero кадры и реалистичные люди	Premium image model	Только после budget gate
PFD Style LoRA	Стиль одного магазина	Open base + style LoRA	Только rights-cleared dataset
PFD Character LoRA	Постоянная героиня	Open base + character LoRA	Identity QA и diversity gate
Manual Route	Высокий правовой или визуальный риск	Выбор редактора	Автопубликация запрещена

7 Контур LoRA обучения

7 Контур LoRA обучения

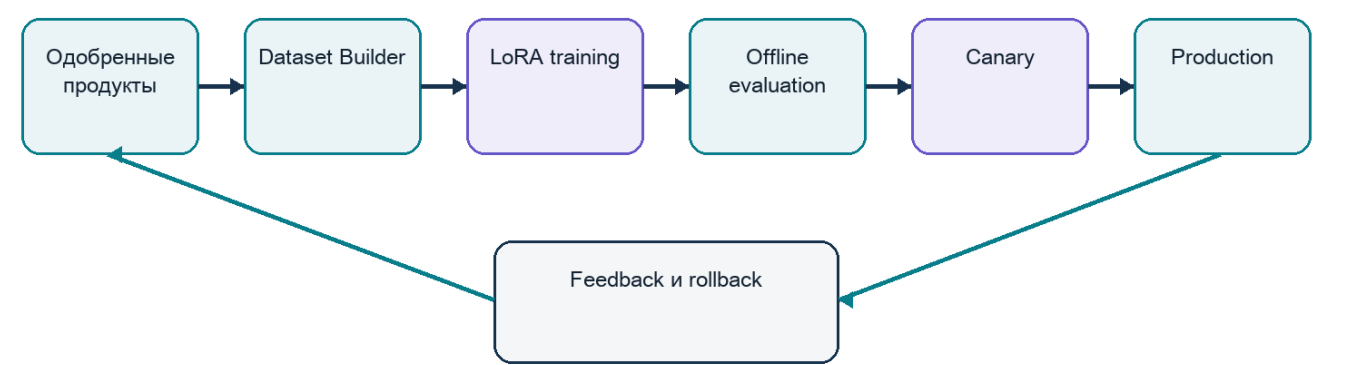


Рисунок 2. Offline обучение, оценка, canary и rollback

LoRA замораживает основные веса базовой модели и обучает небольшой набор дополнительных низкоранговых матриц. Один адаптер должен решать одну задачу: стиль магазина или identity персонажа. Смешивание нескольких магазинов и несовместимых персонажей повышает риск style leakage и потери управляемости.

8 Требования к dataset

- Каждый файл связан с владельцем, источником, лицензией и разрешёнными целями ML использования.
- Удаляются дубликаты, near duplicates, watermark, низкое качество и материалы с неясными правами.
- Captions отделяют постоянные свойства персонажа от одежды, позы, сцены и других изменяемых атрибутов.
- Dataset делится на train, validation и locked test. Locked test не используется в обучении и prompt tuning.
- Версия dataset получает immutable manifest и hashes. Любое изменение создаёт новую версию.
- Изображения людей используются только при документированном согласии на ML обучение.

9 Training run и Model Registry

Параметр	Требование PFD
Base model	Закреплённая версия и проверенная коммерческая лицензия
Configuration	Rank, alpha, learning rate, steps, resolution, seed и precision
Compute	Изолированный GPU worker без production secrets
Artifacts	LoRA weights, logs, sample sheet, metrics и dependency manifest
Reproducibility	Run ID, commit hash, container digest и random seeds
Registry	Статусы candidate, evaluation, canary, production, rollback и retired

10 QA метрики и выпуск

Метрика	Определение	Назначение
First pass rate	Доля продуктов, прошедших QA с первой попытки	Качество и стоимость
Human approval	Принято редактором без регенерации	Основной бизнес KPI
Cost per approved	Все API и GPU расходы на принятый продукт	Контроль бюджета
Novelty pass	Текстовое и визуальное сходство ниже порога	Защита от повторов
Identity consistency	Стабильность лица и ключевых признаков	Character LoRA
Pose diversity	Разные позы, одежда, ракурсы и фон	Защита от overfit
Anatomy defects	Лишние конечности, пальцы и изолированные части тела	Hard block для людей
Text artifacts	Watermark, pseudo text, подписи и fake UI	Hard block если не запрошено

11 Контроль расходов

- Одна платная генерация создаёт master; галерея строится локально.
- Не более двух платных image generation attempts на продукт.
- Fallback не может скрыто выполнять дополнительную дорожную генерацию.
- Стоимость оценивается и резервируется до API call.
- Каждый вызов записывает provider, model, quality, quantity и estimated cost.
- Circuit breaker останавливает pipeline при аномальных retries или расходах.

12 Основные сущности данных

Сущность	Ключевые данные
generation_run	Agent, concept, model versions, cost, status и timestamps
style_profile	Style DNA version, scope, references и forbidden traits
character_profile	Identity, references, consent и allowed variations
feedback_event	Decision, reason, edits, sales, defect и model version
dataset_version	Files, captions, rights, splits, hashes и status
training_run	Base model, LoRA config, GPU, logs, metrics и artifacts
model_registry	Adapter version, compatibility, rollout и rollback target
cost_ledger	Provider operation, quantity, unit cost и actual cost

13 План внедрения

Фаза	Цель	Результат
0	Стабилизировать текущего агента	One master, max 2 attempts, Vision QA, manual review
1	Character Library и rights	References, consent и rights status
2	Retrieval и novelty	Embeddings, vector index, pHash и feedback memory
3	Eval platform	Locked prompts, baseline, blind review и метрики
4	LoRA pilot	Один стиль или один персонаж
5	Hybrid Router	Economy, premium и LoRA profiles
6	Canary и Registry	Monitoring, promotion и rollback
7	Controlled automation	Автопубликация только доверенных классов

14 Критерии готовности первой LoRA модели PFD

- Определена одна узкая задача и владелец бизнес результата.
- Все training assets имеют подтверждённые права на ML обучение.
- Есть versioned dataset с train, validation и locked test.
- Base model и лицензия закреплены в model profile.
- Candidate улучшает целевую метрику и не ухудшает безопасность, новизну и стоимость.
- Blind review и canary успешно завершены.
- Настроены monitoring, kill switch и rollback.

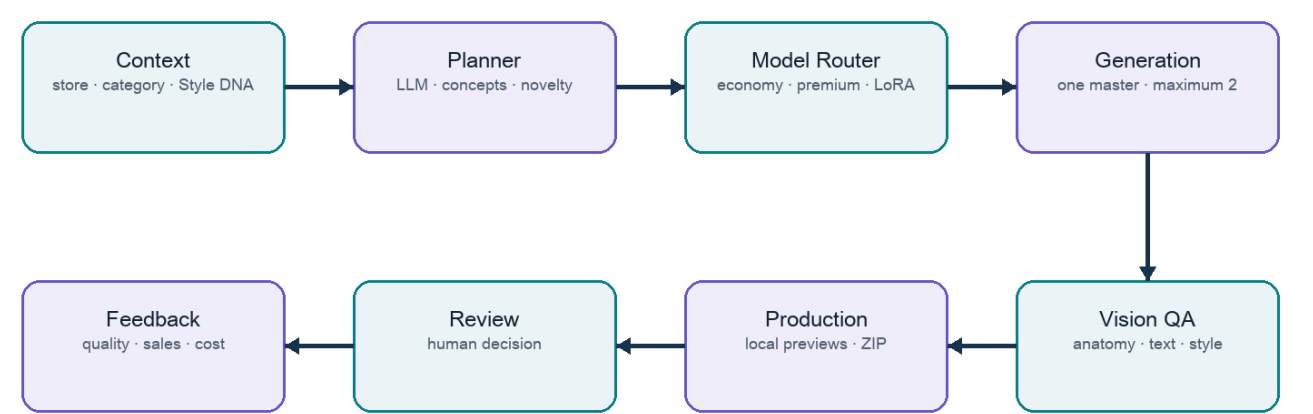
15 Источники

- [LoRA: Low-Rank Adaptation of Large Language Models](#)
- [Hugging Face Diffusers LoRA training documentation](#)
- [Hugging Face Diffusers DreamBooth and FLUX examples](#)
- [OpenAI Image generation API guide](#)
- [OpenAI GPT Image 1 Mini model reference](#)

PFD Hybrid AI Product Generator Architecture

The first integrated LoRA learning and automation loop

2 Hybrid architecture



Feedback feeds PFD memory, datasets and LoRA training; the Model Registry returns approved versions to the Router.

Figure 1. PFD hybrid workflow from context to the Model Registry

This document defines how PicsForDesign builds a hybrid AI product generator. Ready-made text and image models perform base generation, while the PFD control layer manages context, memory, novelty, quality, cost and LoRA training.

1 Core architecture decision

PFD will not train a foundation model from scratch. The first version uses external models through APIs and a PFD-owned orchestrator. After a licensed and reviewed dataset has been accumulated, PFD trains compact LoRA adapters for specific styles, stores and persistent characters. A PFD-trained model is the combination of a LoRA adapter version, dataset version, model profile and approved evaluation suite.

3 Adaptation and learning levels

Level	Mechanism	Weight training	Purpose	GPU
A	Prompt rules and structured context	No	Format, style, constraints and tone of voice	No
B	RAG, embeddings, references and control adapters	No	Memory, retrieval, composition and identity guidance	Optional
C	LoRA fine tuning	Yes, small adapter	Persistent store style or character identity	For training

4 Current and target state

Area	Current	Target
Generation	One master with gpt-image-1-mini low	Router across economy, premium and LoRA
Cost	No more than two paid attempts	Budget ledger and pre-call blocking
Previews	Local derivatives from the master	Templates by product type and channel
QA	Vision QA and visual uniqueness gate	QA profiles for people, characters and creative kits
Memory	Archive memory and metadata Style DNA	Vector retrieval, Character Library and feedback memory
Training	Not performed	Versioned LoRA training and Model Registry

5 End-to-end product generation workflow

Step	Stage	Algorithm	Output and gate
01	Trigger	Schedule, manual action or API creates a run with an idempotency key	Exactly one task
02	Budget	Daily, per-product and credit limits are checked	Allowed or blocked_budget
03	Rights	Archive, character and training asset rights are checked	Allowed or blocked_rights
04	Context	Store, category, audience, language, format and channel	Context snapshot
05	Style DNA	Palette, themes, vocabulary, composition and forbidden traits	Versioned style brief
06	Character Profile	Identity references, clothing, age and allowed variation	Character brief or none
07	Memory	Vector retrieval selects diverse works and feedback events	Reference set
08	Concepts	The LLM creates at least three materially different ideas	Concept candidates
09	Text novelty	Title, description, tags and concept embeddings are compared	Novelty score
10	Selection	A ranker balances relevance, novelty, risk and budget	Selected concept
11	Prompt	Prompt, negative constraints, QA rubric and outputs are built	Prompt package
12	Router	Economy, premium or LoRA profile is selected	Model profile
13	Master	One image is generated with n = 1	Master candidate
14	Safety	Format, prohibited content, text and watermark are checked	Pass or reject
15	Vision QA	Anatomy, face, style, composition and element integrity	QA report
16	Image novelty	Image embedding and perceptual hash are compared to the archive	Visual novelty score
17	Correction	One final targeted repair attempt	Second candidate or blocked
18	Derivatives	Thumbnail, gallery and marketplace sizes are built locally	No extra image API cost
19	Metadata	Title, description, tags, license and manifest	Draft package
20	Human review	An editor approves, rejects or edits with a reason code	Decision event
21	Publishing	The approved package receives a product ID	Published product
22	Feedback	Decision, sales, defects, edits and cost are recorded	Training evidence
23	Dataset	Only rights-cleared reviewed examples enter a new version	Candidate dataset
24	LoRA cycle	Training, offline evaluation, canary, promotion or rollback	Approved adapter

6 Model Router

The router does not choose the newest model. It chooses the least expensive profile capable of meeting the quality threshold. Uncertain or high-risk tasks are routed to manual review.

Profile	Use case	Model	Constraint
Draft Economy	Ideas and low-risk creative kits	Low-cost API image model	One master, at most 2 attempts
Premium Master	Hero images and realistic people	Premium image model	Only after the budget gate
PFD Style LoRA	One store's visual style	Open base + style LoRA	Rights-cleared dataset only
PFD Character LoRA	A persistent character	Open base + character LoRA	Identity QA and diversity gate
Manual Route	High legal or visual risk	Editor-selected pipeline	No automatic publishing

7 LoRA training loop

7 LoRA training loop

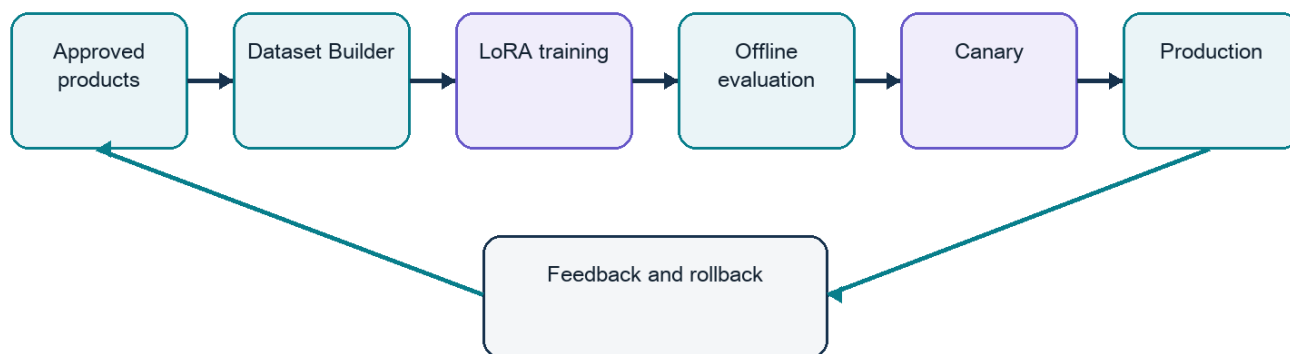


Figure 2. Offline training, evaluation, canary and rollback

LoRA freezes the main weights of a base model and trains a small set of added low-rank matrices. A single adapter should solve one clear task: a store style or a character identity. Mixing unrelated stores and characters increases style leakage and reduces controllability.

8 Dataset requirements

- Every file is linked to an owner, source, license and permitted machine-learning uses.
- Duplicates, near duplicates, watermarks, low-quality assets and unclear rights are removed.
- Captions separate persistent identity traits from clothing, pose, scene and other variable attributes.
- The dataset is split into train, validation and locked test. Locked test is excluded from training and prompt tuning.
- Each dataset version receives an immutable manifest and hashes. Any change creates a new version.
- Images of people require documented consent for machine-learning training.

9 Training run and Model Registry

Parameter	PFD requirement
Base model	Pinned version and reviewed commercial license
Configuration	Rank, alpha, learning rate, steps, resolution, seed and precision
Compute	Isolated GPU worker without production secrets
Artifacts	LoRA weights, logs, sample sheet, metrics and dependency manifest
Reproducibility	Run ID, commit hash, container digest and random seeds
Registry	Candidate, evaluation, canary, production, rollback and retired states

10 Quality metrics and release

Metric	Definition	Purpose
First-pass rate	Share of products passing QA on the first attempt	Quality and cost
Human approval	Accepted without regeneration	Primary business KPI
Cost per approved	All API and GPU cost per approved product	Budget control
Novelty pass	Text and image similarity below thresholds	Duplicate prevention
Identity consistency	Stable face and key character traits	Character LoRA
Pose diversity	Varied poses, clothing, angles and backgrounds	Overfitting protection
Anatomy defects	Extra limbs, fingers and isolated body parts	Hard block for people
Text artifacts	Watermarks, pseudo-text, signatures and fake UI	Hard block unless requested

11 Cost safeguards

- One paid generation creates the master; the gallery is built locally.
- No more than two paid image-generation attempts per product.
- Fallbacks cannot silently trigger an additional expensive generation.
- Cost is estimated and reserved before an API call.
- Every call records provider, model, quality, quantity and estimated cost.
- A circuit breaker stops the pipeline when retries or spending become abnormal.

12 Core data entities

Entity	Key data
generation_run	Agent, concept, model versions, cost, status and timestamps
style_profile	Style DNA version, scope, references and forbidden traits
character_profile	Identity, references, consent and allowed variation
feedback_event	Decision, reason, edits, sales, defects and model version
dataset_version	Files, captions, rights, splits, hashes and status
training_run	Base model, LoRA config, GPU, logs, metrics and artifacts
model_registry	Adapter version, compatibility, rollout and rollback target
cost_ledger	Provider operation, quantity, unit cost and actual cost

13 Implementation roadmap

Phase	Goal	Deliverable
0	Stabilize the current agent	One master, max 2 attempts, Vision QA and manual review
1	Character Library and rights	References, consent and rights status
2	Retrieval and novelty	Embeddings, vector index, pHash and feedback memory
3	Evaluation platform	Locked prompts, baseline, blind review and metrics
4	LoRA pilot	One style or one character
5	Hybrid Router	Economy, premium and LoRA profiles
6	Canary and Registry	Monitoring, promotion and rollback
7	Controlled automation	Automatic publishing for trusted classes only

14 Readiness criteria for the first PFD LoRA model

- One narrow adapter task and business owner are defined.
- Every training asset has confirmed machine-learning rights.
- A versioned dataset contains train, validation and locked test splits.
- The base model and license are pinned in the model profile.
- The candidate improves the target metric without degrading safety, novelty or cost.
- Blind review and canary deployment have passed.
- Monitoring, kill switch and rollback are configured.

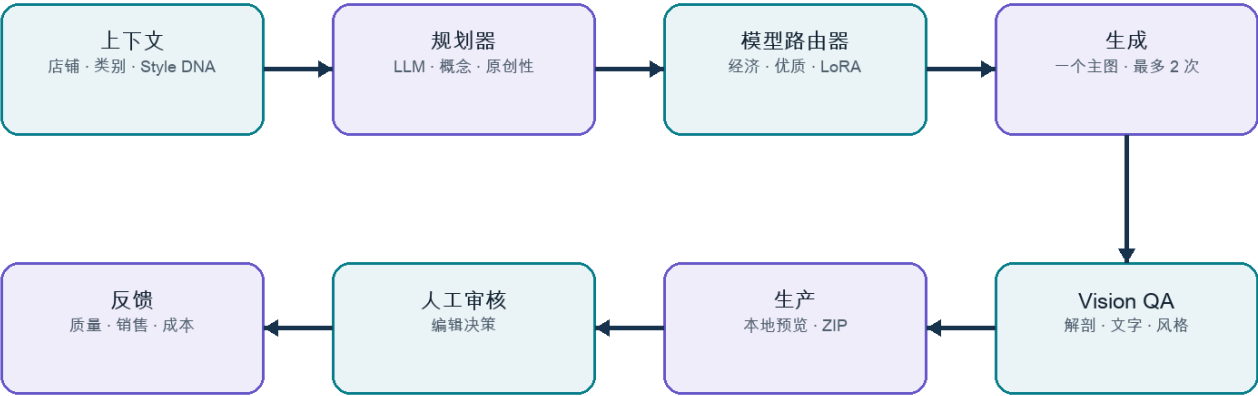
15 References

- [LoRA: Low-Rank Adaptation of Large Language Models](#)
- [Hugging Face Diffusers LoRA training documentation](#)
- [Hugging Face Diffusers DreamBooth and FLUX examples](#)
- [OpenAI Image generation API guide](#)
- [OpenAI GPT Image 1 Mini model reference](#)

PFD 混合式 AI 产品生成器架构

首个集成 LoRA 学习与自动化闭环

2 混合式架构



反馈进入 PFD 记忆、数据集与 LoRA 训练；模型注册表将批准版本返回路由器。

图 1 PFD 从上下文到模型注册表的混合工作流

本文档定义 PicsForDesign 如何构建混合式 AI 产品生成器。现成的文本与图像模型负责基础生成，PFD 自有控制层负责上下文、记忆、原创性、质量、成本以及 LoRA 训练。

1 核心架构决策

PFD 不从零训练基础模型。第一版通过 API 使用外部模型，并由 PFD 自有编排器控制。在积累经过授权和审核的数据集后，PFD 为特定风格、店铺和固定角色训练轻量 LoRA 适配器。PFD 的已训练模型由 LoRA 适配器版本、数据集版本、模型配置文件和已批准评测集共同构成。

3 适配与学习层级

层级	机制	训练权重	用途	GPU
A	提示词规则与结构化上下文	否	格式、风格、限制与语气	否
B	RAG、向量嵌入、参考图与控制适配器	否	记忆、检索、构图与角色身份控制	可选
C	LoRA 微调	是，轻量适配器	固化店铺风格或固定角色身份	训练时需要

4 当前状态与目标状态

领域	当前	目标
生成	使用 gpt-image-1-mini low 生成一个主图	在经济、优质与 LoRA 模型之间路由
成本	最多两次付费尝试	成本台账与调用前预算阻断
预览	由主图在本地生成衍生图	按产品类型和渠道使用模板
质量	Vision QA 与视觉原创性门控	人物、角色与创意套件的专用 QA 配置
记忆	档案记忆与元数据 Style DNA	向量检索、角色库与反馈记忆
训练	尚未执行	版本化 LoRA 训练与模型注册表

5 端到端产品生成 workflow

步骤	阶段	算法	输出与门控
01	触发	计划任务、人工操作或 API 以幂等键创建运行	只创建一个任务
02	预算	检查每日、单产品与积分额度	允许或 blocked_budget
03	权利	检查档案、角色与训练素材权利	允许或 blocked_rights
04	上下文	店铺、类别、受众、语言、格式与渠道	上下文快照
05	Style DNA	色彩、主题、词汇、构图与禁止特征	版本化风格说明
06	角色配置	身份参考、服装、年龄与允许变化	角色说明或无
07	记忆	向量检索选择多样化作品与反馈事件	参考集合
08	概念	LLM 创建至少三个实质不同的创意	候选概念
09	文本原创性	比较标题、描述、标签与概念向量	原创性分数
10	选择	排序器平衡相关性、原创性、风险与预算	选定概念
11	提示包	生成提示词、负面限制、QA 标准与预期输出	Prompt package
12	路由	选择经济、优质或 LoRA 配置	模型配置
13	主图	仅生成一张图 n = 1	主图候选
14	安全	检查格式、禁止内容、文字与水印	通过或拒绝
15	Vision QA	检查解剖、面部、风格、构图与元素完整性	QA 报告
16	图像原创性	图像向量与感知哈希和档案比较	视觉原创性分数
17	修复	只允许一次有针对性的最终修复	第二候选或阻断
18	衍生图	在本地生成缩略图、图库图与市场尺寸	不增加图像 API 成本
19	元数据	标题、描述、标签、许可与清单	草稿包
20	人工审核	编辑以原因代码批准、拒绝或修改	决策事件
21	发布	批准的产品包获得产品 ID	已发布产品
22	反馈	记录决策、销售、缺陷、修改与成本	训练证据
23	数据集	只有权利清晰且已审核的样本进入新版本	候选数据集
24	LoRA 周期	训练、离线评测、灰度、晋级或回滚	批准的适配器

6 模型路由器

路由器不选择最新模型，而选择能够达到质量阈值的最低成本配置。不确定或高风险任务进入人工审核。

配置	使用场景	模型	限制
Draft Economy	创意草稿与低风险创意套件	低成本 API 图像模型	一个主图，最多 2 次
Premium Master	主视觉与写实人物	优质图像模型	通过预算门控后使用
PFD Style LoRA	单一店铺视觉风格	开放基础模型 + 风格 LoRA	仅授权数据集
PFD Character LoRA	固定角色	开放基础模型 + 角色 LoRA	身份 QA 与多样性门控
Manual Route	高法律或视觉风险	编辑选择流程	禁止自动发布

7 LoRA 训练闭环

7 LoRA 训练闭环

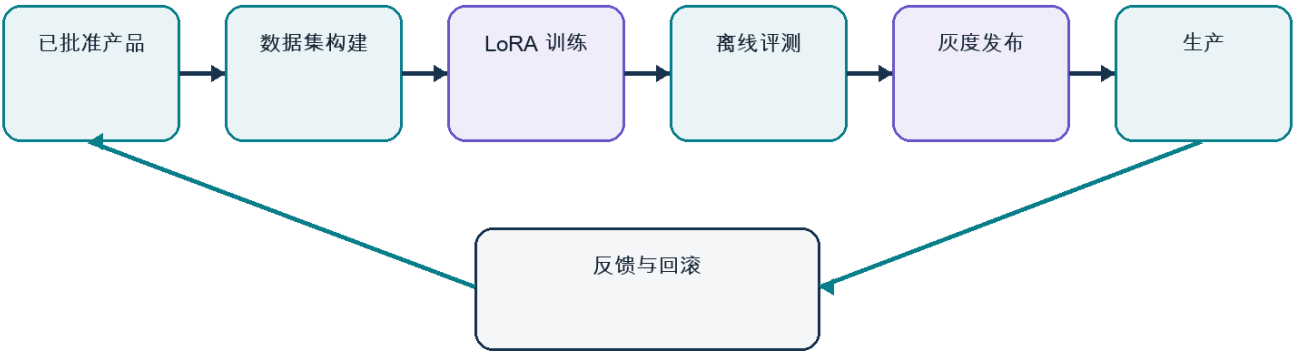


图 2 离线训练、评测、灰度发布与回滚

LoRA 冻结基础模型的主要权重，只训练一组新增的低秩矩阵。一个适配器应解决一个明确任务：店铺风格或角色身份。将无关店铺和角色混入同一适配器会增加风格泄漏并降低可控性。

8 数据集要求

- 每个文件都关联所有者、来源、许可及允许的机器学习用途。
- 删除重复、近重复、水印、低质量素材和权利不清晰的内容。
- 标注应区分固定身份特征与服装、姿势、场景等可变属性。
- 数据集划分为训练、验证和锁定测试集。锁定测试集不得用于训练或提示词调优。
- 每个数据集版本包含不可变清单与哈希。任何修改都会产生新版本。
- 人物图像必须具备明确的机器学习训练同意。

9 训练运行与模型注册表

参数	PFD 要求
基础模型	固定版本并审查商业许可
配置	Rank、alpha、学习率、步数、分辨率、seed 与 precision
计算	隔离 GPU worker，不持有 production secrets
产物	LoRA 权重、日志、样例表、指标与依赖清单
可复现性	Run ID、commit hash、容器摘要与随机种子
注册状态	candidate、evaluation、canary、production、rollback、retired

10 质量指标与发布

指标	定义	用途
首轮通过率	第一次生成即通过 QA 的产品比例	质量与成本
人工批准率	无需重新生成即可被编辑接受	核心业务 KPI
单个批准产品成本	每个批准产品的全部 API 与 GPU 成本	预算控制
原创性通过率	文本与图像相似度低于阈值	防止重复
身份一致性	面部与关键角色特征稳定	Character LoRA
姿势多样性	姿势、服装、角度与背景多样	防止过拟合
解剖缺陷	多余肢体、手指与孤立身体部位	人物任务硬阻断
文字伪影	水印、伪文字、签名与假 UI	非需求时硬阻断

11 成本保护

- 一次付费生成只创建主图，图库图在本地生成。
- 每个产品最多两次付费图像生成。
- Fallback 不得隐藏触发额外昂贵生成。
- 在 API 调用前估算并预留成本。
- 每次调用记录 provider、model、quality、quantity 与 estimated cost。
- 当重试或支出异常时，熔断器停止流程。

12 核心数据实体

实体	关键数据
generation_run	Agent、concept、模型版本、成本、状态与时间
style_profile	Style DNA 版本、范围、参考与禁止特征
character_profile	身份、参考、同意与允许变化
feedback_event	决策、原因、修改、销售、缺陷与模型版本
dataset_version	文件、标注、权利、划分、哈希与状态
training_run	基础模型、LoRA 配置、GPU、日志、指标与产物
model_registry	适配器版本、兼容性、发布与回滚目标
cost_ledger	Provider 操作、数量、单价与实际成本

13 实施路线图

阶段	目标	交付物
0	稳定现有 Agent	一个主图、最多 2 次、Vision QA 与人工审核
1	角色库与权利	参考图、同意与权利状态
2	检索与原创性	Embeddings、向量索引、pHash 与反馈记忆
3	评测平台	锁定提示集、baseline、盲审与指标
4	LoRA 试点	一个风格或一个角色
5	混合路由	Economy、premium 与 LoRA 配置
6	灰度与注册表	监控、晋级与回滚
7	受控自动化	仅可信类别允许自动发布

14 PFD 首个 LoRA 模型就绪标准

- 定义一个明确的适配器任务与业务负责人。
- 所有训练素材均有明确的机器学习使用权。
- 版本化数据集包含训练、验证与锁定测试划分。
- 基础模型及其许可固定在模型配置文件中。
- 候选模型提升目标指标，并且不降低安全、原创性和成本指标。
- 盲审与灰度发布均已通过。
- 监控、停止开关与回滚均已配置。

15 参考资料

- [LoRA: Low-Rank Adaptation of Large Language Models](#)
- [Hugging Face Diffusers LoRA training documentation](#)
- [Hugging Face Diffusers DreamBooth and FLUX examples](#)
- [OpenAI Image generation API guide](#)
- [OpenAI GPT Image 1 Mini model reference](#)