

**ОБУЧЕНИЕ
МОДЕЛИ**

LORA

для

**STABLE
DIFFUSION**

Онлайн или на своем ПК

СОДЕРЖАНИЕ

I. Введение

Что такое Stable Diffusion? 2-3

Что такое LORA? 2-5

Как это может помочь в айдентике 6-7

II. Онлайн тренировка

О сервисе Scenario 9

Регистрация и обзор интерфейса 10-13

Подготовка датасета 14-16

Тренировка модели и генерация 17-27

II. Тренировка на своем ПК

Загрузка нужного ПО 29-31

Обзор интерфейса и настроек 32-39

Подготовка датасета 40

Тренировка модели и генерация 41-42

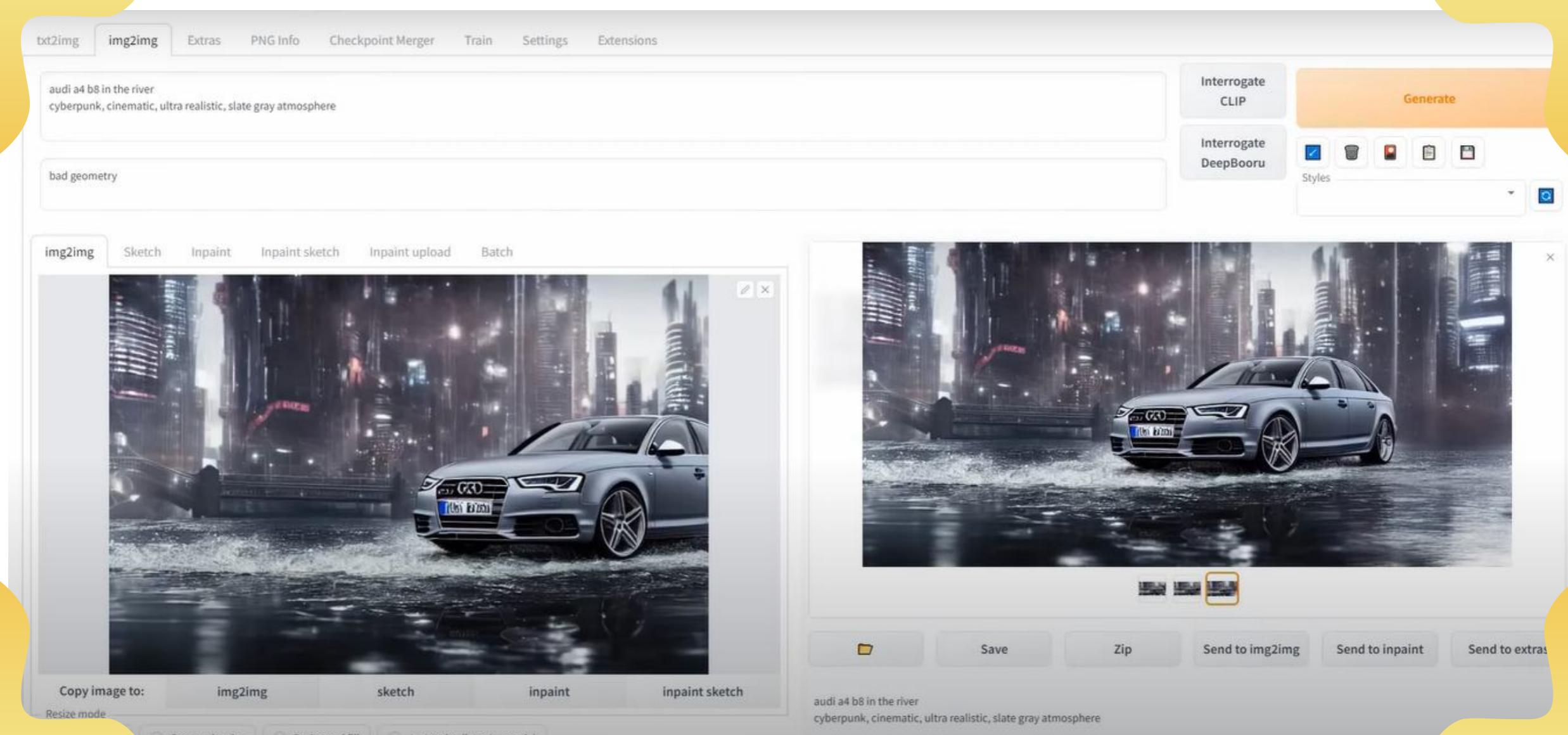


ВВЕДЕНИЕ

ЧТО ТАКОЕ STABLE DIFFUSION?

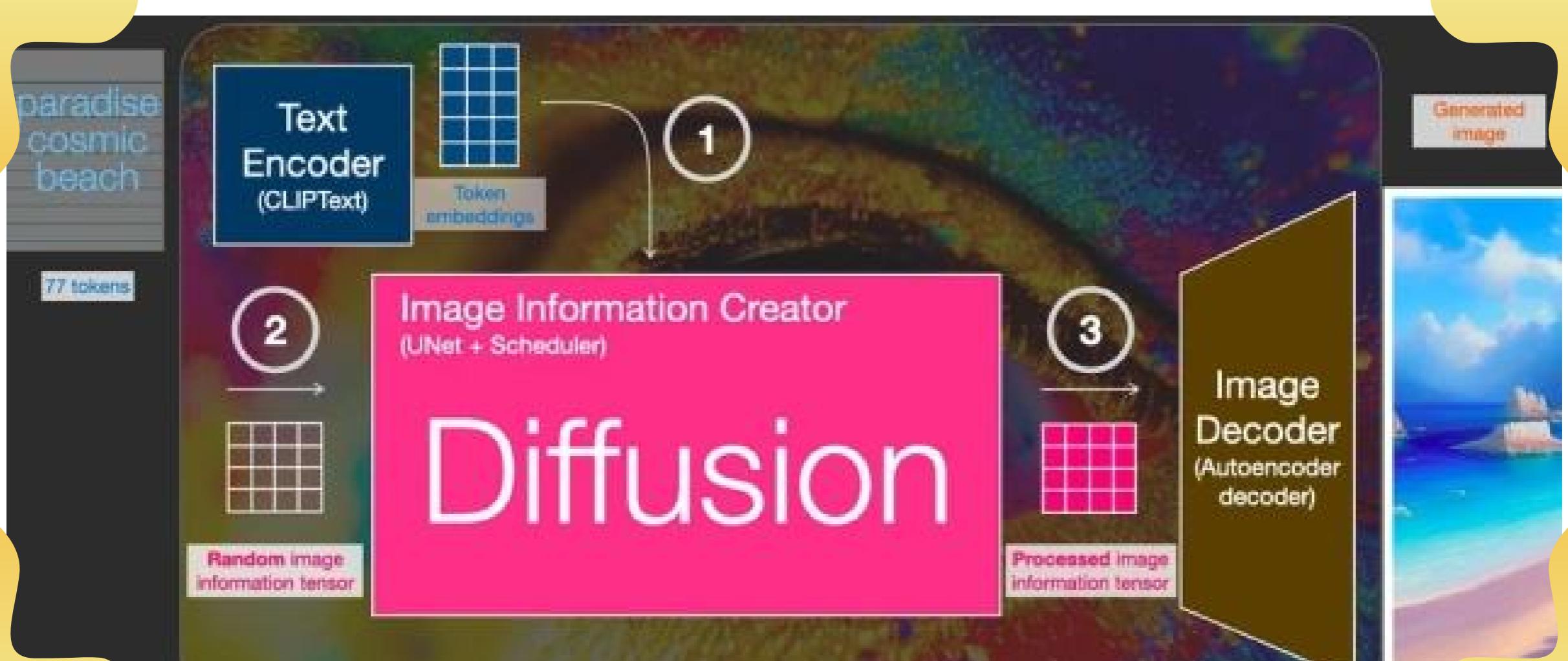
В 2024 году нейросети стали настоящим трендом в работе специалистов разных направлений деятельности, а также и среди обычных пользователей. Обученные алгоритмы способны создавать уникальные изображения, генерировать креативные концепты по текстовым запросам, писать программный код, а также формулировать развернутые ответы на вопросы.

Stable Diffusion (в дальнейшем **SD**), разработанная группой студии Stability.ai с открытым исходным кодом, представляет собой инновационную нейросеть, способную генерировать изображения на основе текстовых запросов и дополнять существующие наброски или редактировать исходные изображения.



SD состоит из нескольких ключевых моделей:

- 1. Кодировщик текста:** это специальная языковая модель-трансформер, которая принимает текст на входе и преобразует его в числовое представление, описывающее каждое слово.
- 2. Генератор изображений:** это комбинация нейросети UNet и алгоритма планирования, который создает информацию об изображении.
- 3. Декодер:** он используется для визуализации изображения на основе полученной информации.

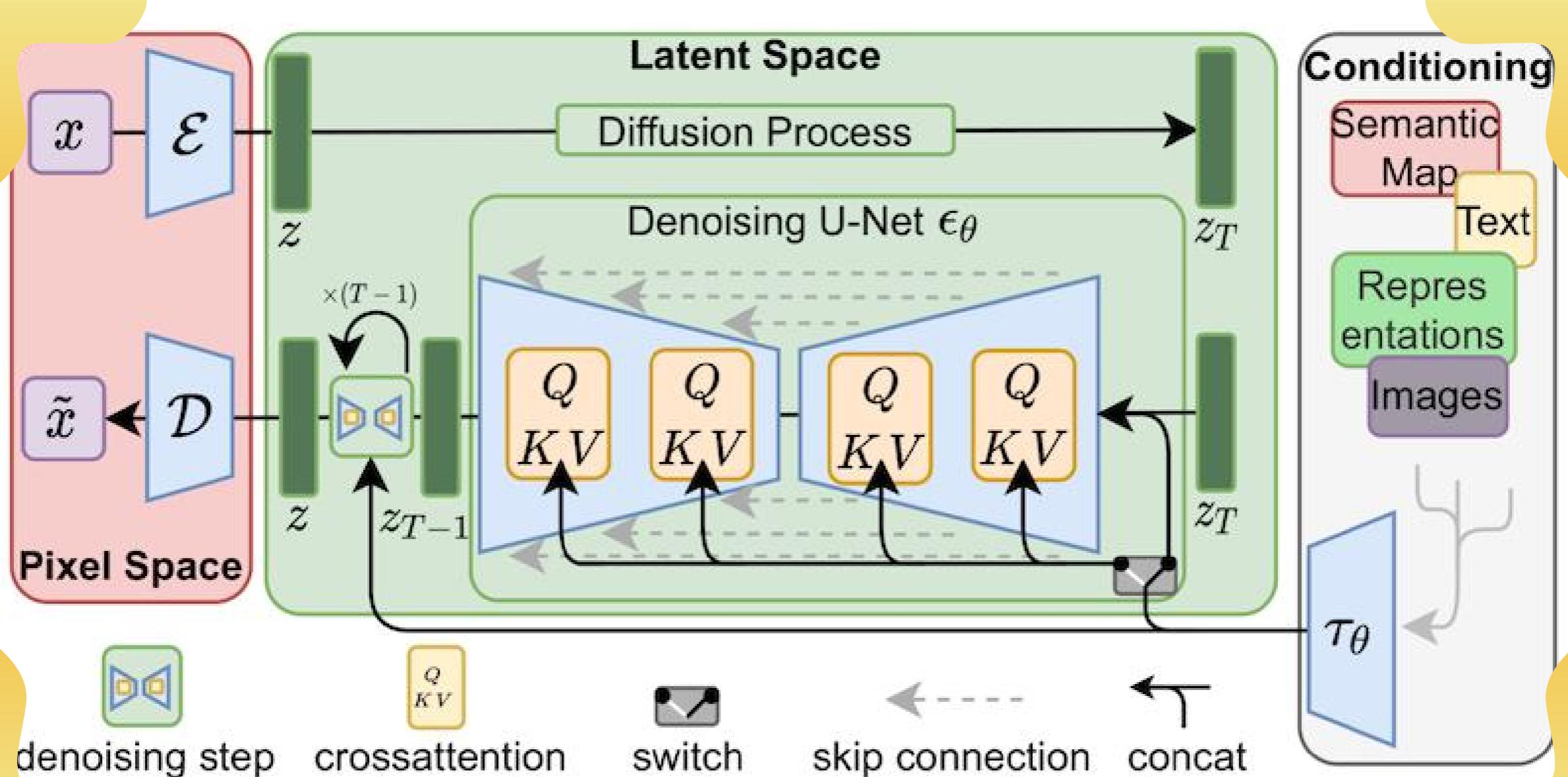


ЧТО ТАКОЕ LORA?

Если пользователю нужен метод, который сбалансирован между размером модели и её обучающей мощностью, то на помощь приходят модели LoRA (Low-Rank Adaptation) для SD. Они отлично подходят для этой задачи. Они компактны, но в то же время эффективны, что делает их превосходным выбором для тех, кто хочет иметь обширную коллекцию моделей, не создавая проблем с дисковым пространством.

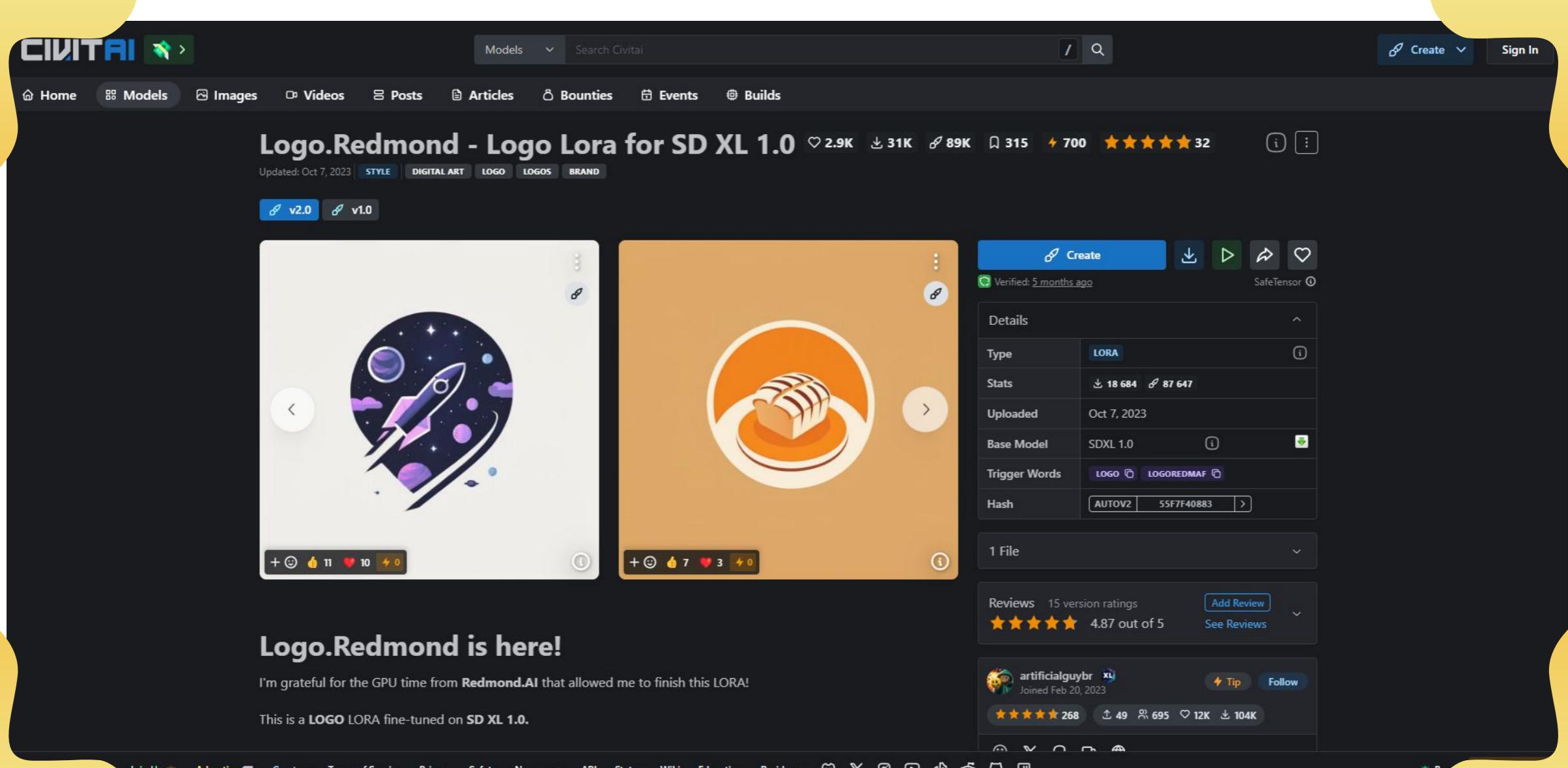


Модели Stable Diffusion LoRA применяет мелкие изменения к стандартным моделям контрольной точки(checkpoint), в результате чего файлы становятся на 10-100 раз меньше по размеру, чем у своих аналогов. При этом модели LoRA не понижают обучающие возможности, сохраняя баланс баланса между размером файла и возможностями тонкой настройки. Модели LoRA работают путем настройки слоев кросс-внимания моделей Stable Diffusion, то есть той области, где пересекаются изображение и подсказка. Выяснилось, что тонкая настройка этой части модели достаточна для получения желаемых результатов обучения.



ЗАЧЕМ В АЙДЕНТИКЕ?

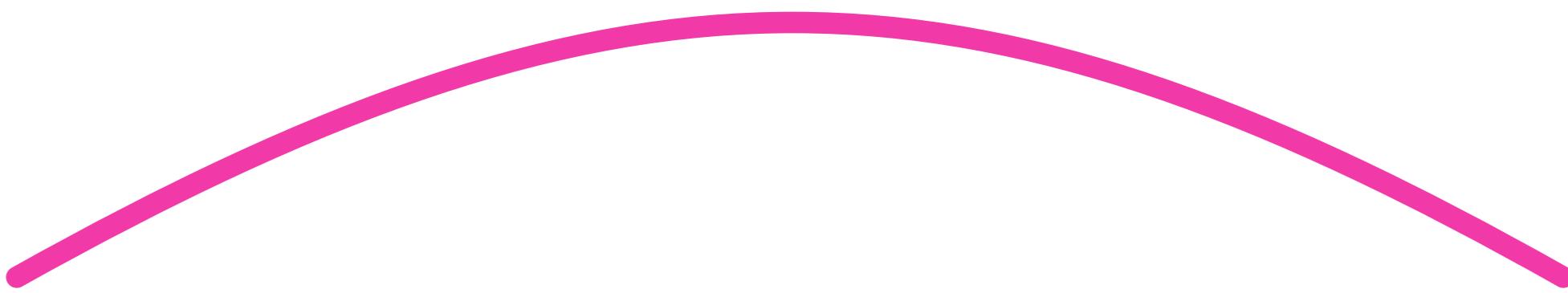
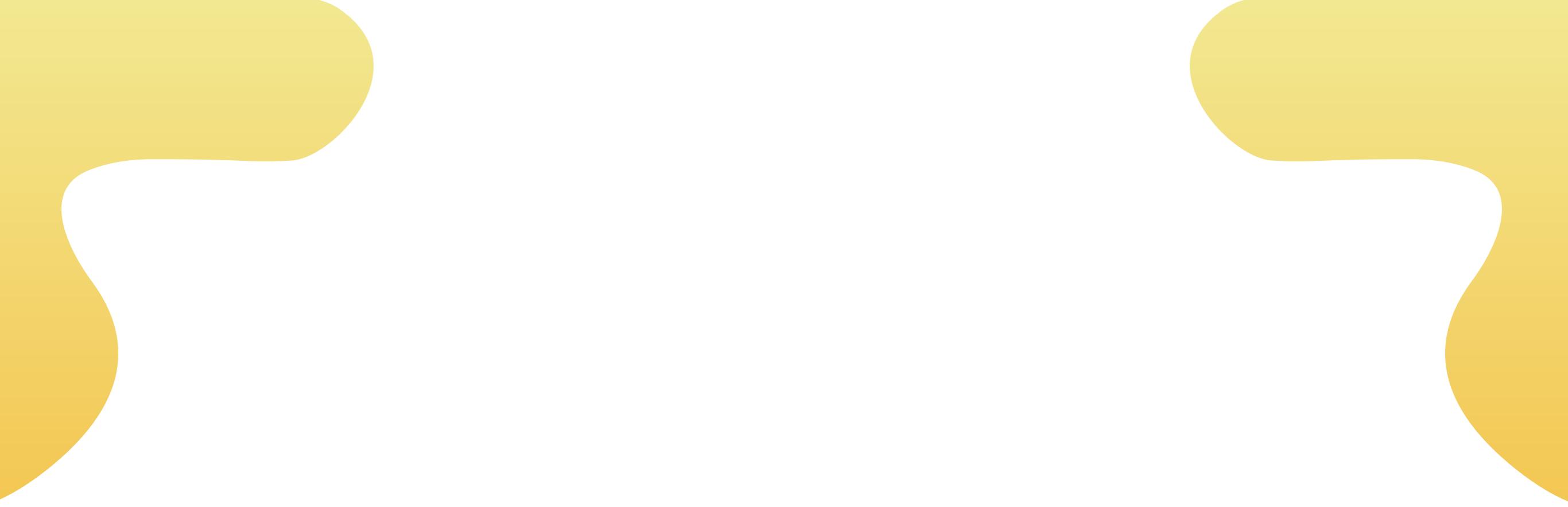
С помощью моделей LoRA можно генерировать разнообразные визуальные элементы, такие как логотипы, анимационных персонажей, иллюстрации для рекламных материалов и социальных сетей, а также уникальные фоны и текстуры для веб-сайтов и приложений. Благодаря гибкости и возможности тонкой настройки моделей LoRA, можно создавать изображения, которые точно передают корпоративную идентичность и максимально привлекают внимание аудитории.



Кроме того, генерация изображений для бренда с использованием моделей LoRA может существенно сэкономить время и ресурсы, позволяя быстро получать качественный визуальный контент без необходимости привлечения дорогостоящих студий или дизайнеров. Это делает модели LoRA наиболее интересным вариантом для использования в сфере малого и среднего бизнеса, а также стартапов, которым важно создание качественного и профессионального визуального контента с минимальными затратами ресурсов.

В этом руководстве обучение LoRA будет показано на примере для маскота ППОС ИГУ.





ТРЕНИРОВКА

LORA

ОНЛАЙН

Подходит для новичков

О СЕРВИСЕ SCENARIO

Scenario (app.scenario.com) - это передовое приложение искусственного интеллекта для создания изображений из текста, которое позиционирует себя как сервис для разработчиков игр. Оно предоставляет пользователям контроль над генерацией концепций благодаря своим возможностям тонкой настройки на основе Stable Diffusion. Помимо веб-приложения, Scenario расширяет свой охват на мобильные приложения для iOS и Android, бота в Discord, плагин для Unity и API для индивидуальных творческих проектов.

Scenario предлагает такие возможности, как:

- Обучение собственных моделей;
- Генерация изображений с различными настройками с помощью готовых моделей;
- Инструментарий для редактирования сгенерированных изображений, апскейла, пикселизации;
- Организация рабочего пространства для команд и многое другое.



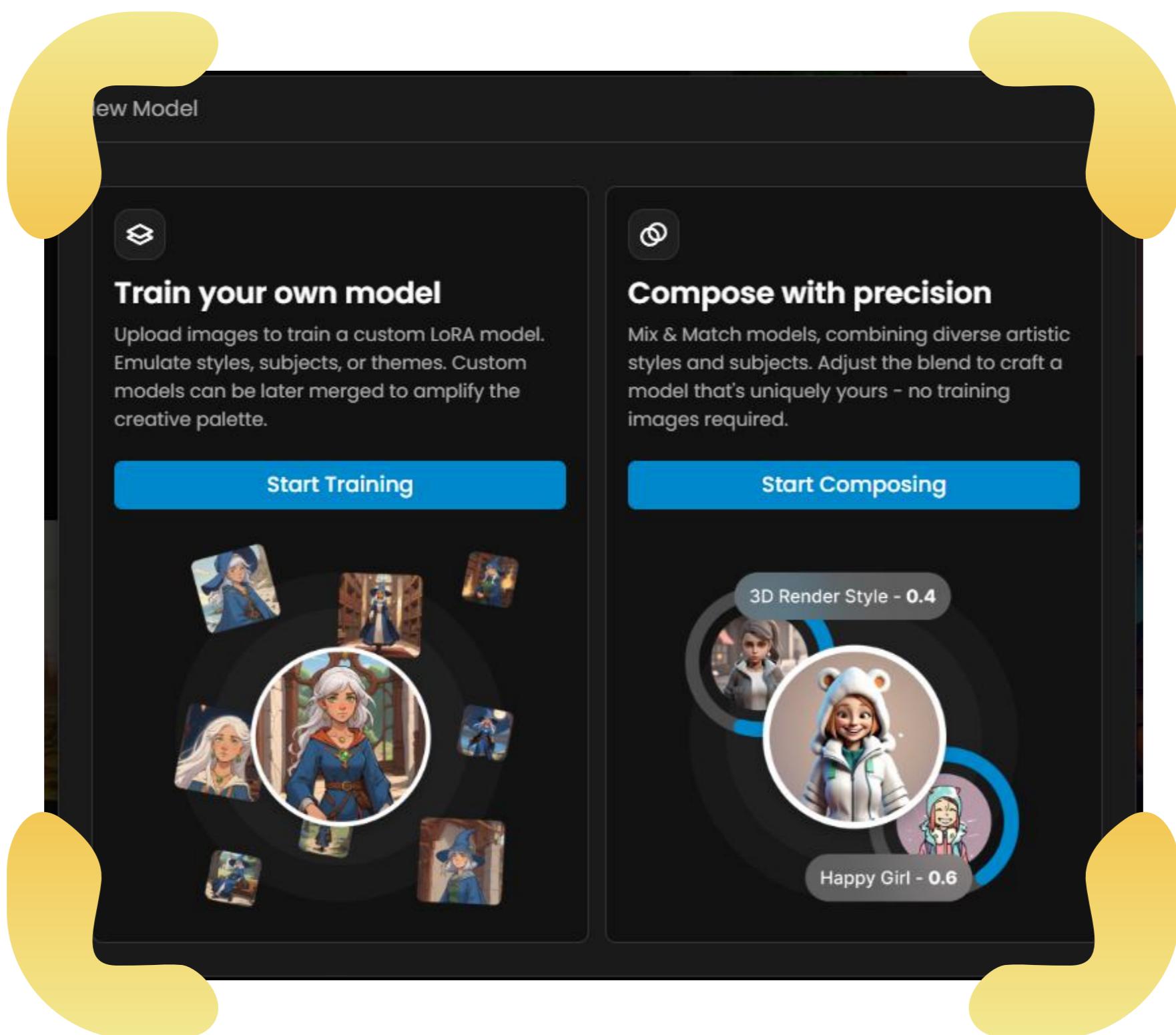
Все вышеперечисленные функции (кроме совместного пространства) доступны бесплатно.



ОБЗОР ИНТЕРФЕЙСА

После регистрации на сервисе пользователь попадает на главный экран, на котором сразу видно, какие функции предлагает сервис.

Можно просмотреть каждый инструмент, но нас интересует вкладка Models, поэтому нажимаем туда на плюсик.



Появляется окно, в котором видим, что кроме тренировки собственной модели, есть возможность совместить модели Lora с собственными или моделями от сервиса. Нам нужно сделать свою "чистую" LoRA, поэтому для наших нужд понадобится кнопка Start Training.

Появляется страничка с настройками тренировки и выбором датасета.

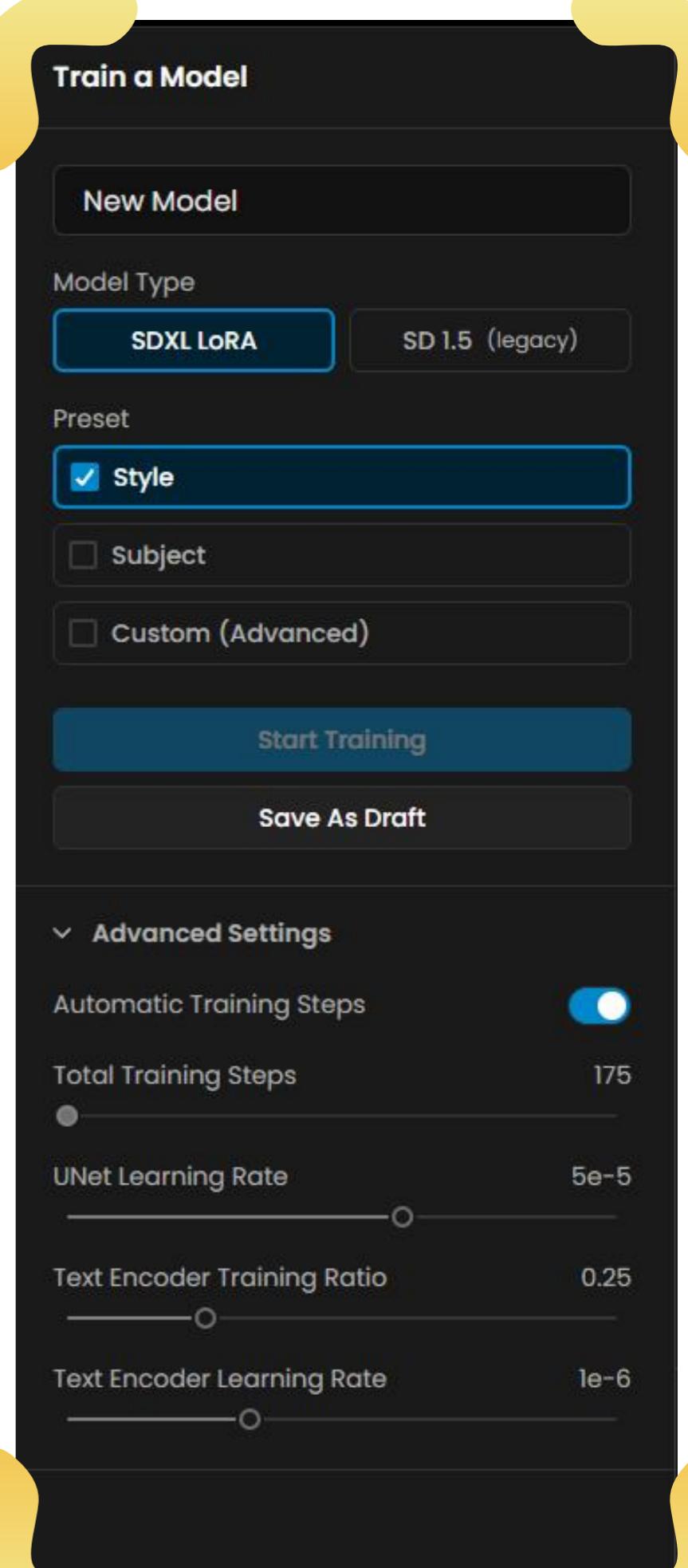
По настройкам:

Model Type: предпочтительнее SDXL, так как судя по экспериментам пользователей, эта модель лучше справляется со стилем, она более совершенна и новее. Но и обучение будет проходить дольше.

Preset: Если не хотите играть с настройками, то оставляем Style. Для тренировки на предметах - Subject.

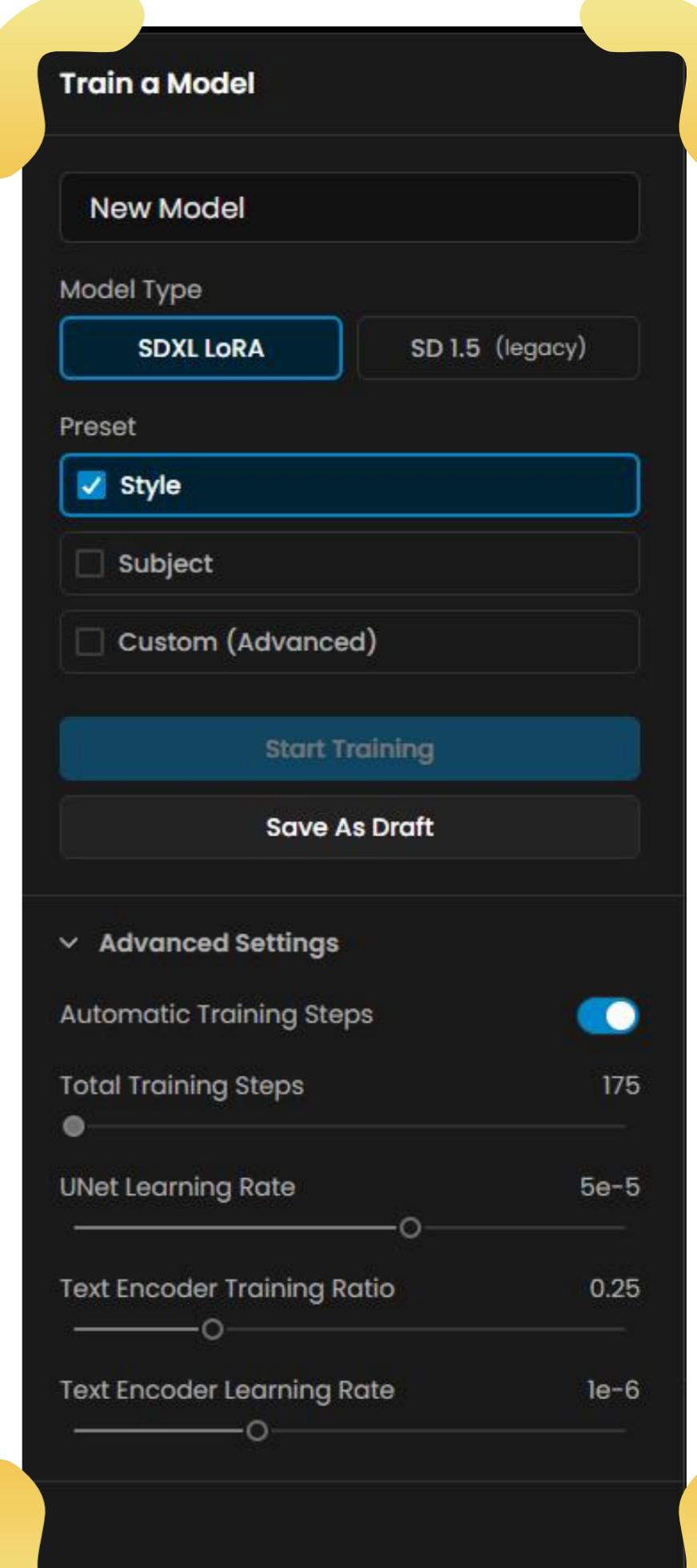
Total Training Steps: количество раз, которые нейросеть проходит через обучающие данные. Увеличение этого числа позволяет сети лучше настраиваться на данные, что может улучшить качество модели, а может сделать так, что она переучится.

Лучше оставить ползунок Automatic включенным.



UNet Learning Rate: что-то вроде памяти, имеет информацию о том как элементы взаимодействуют друг с другом. Первое куда стоит смотреть если обучение не получилось. Сначала рекомендую пройтись со стандартными параметрами и что-то крутить только когда вроде все должно было получиться, но не получилось. Если итоговая модель выдает визуальный шум вместо нормального изображения то значение слишком высокое. попробуйте слегка уменьшить и повторить.

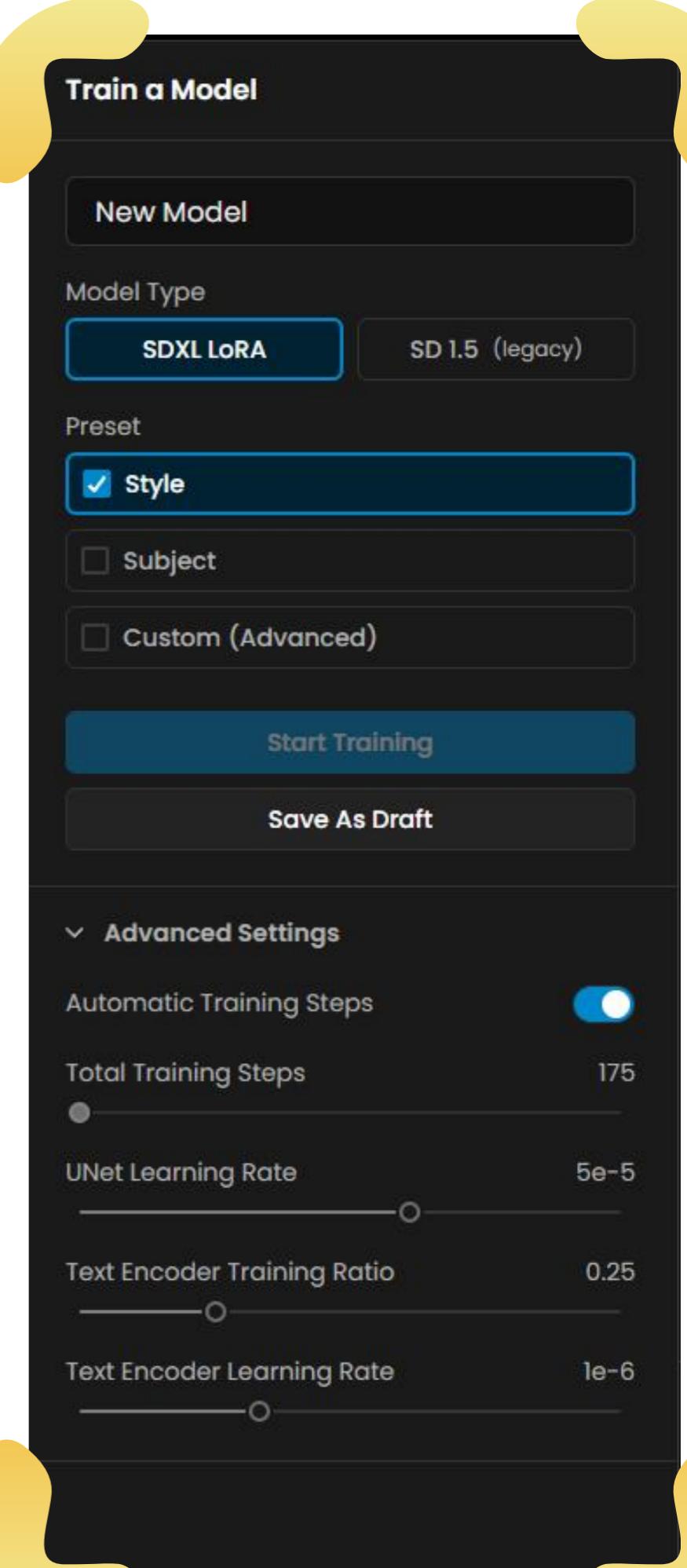
Text Encoder Training Ratio: Определяет, насколько важно сравнивать текстовые описания с изображениями в процессе обучения. Чем выше этот коэффициент, тем сильнее учитывается текстовая информация при обучении. Это может привести к более точному соответствуанию между описанием и изображением.



Text Encoder Training Rate: это довольно сложная настройка. Меньшие значения могут создавать более странные, но иногда интересные результаты. С другой стороны, большие значения могут показывать более гармоничное сочетание с вторым запросом.

Save As Draft: нажав на эту кнопку, не до конца заполненный шаблон настроек сохраняется во вкладку Models. В любое время можно его отредактировать, изменить/ добавить фотографии позже.

С настройками всё. Если что-то получится некрасиво, можно поэкспериментировать с ними. Осталось разобраться с датасетом, на котором будем обучать модель.



ПОДГОТОВКА ДАТАСЕТА

Если тренируете LoRA на основе SD 1.5, то для этой версии нужны изображения с разрешением 512 x 512, так как эта модель Stable Diffusion обучена на таком разрешении. Если обучаете на высоком разрешении, то выбираем SDXL, там 1024 x 1024.

Все ошметки чужих конечностей, непонятные объекты на изображении, подписи художников, ссылки – все, что не хотите видеть в генерациях, лучше обрезать или замазать в редакторе фотографий.

В попытках добрать нужное количество изображений не стоит брать некачественные, это может наоборот ухудшить результат. Лучше пусть будет меньше, но хороших. Так же важный момент в том, что на обучение можно отдавать фотографии сгенерированные самой нейросетью. Если есть персонаж которого хотите “закрепить”, это отличный способ сделать это. Или натренировали модель, но она выдает хорошие результаты редко. Берете удачные и тоже отправляете в датасет.



Если тренируете персонажа/человека, как в далее рассматриваемом примере, то желательно чтобы датасет состоял только из изображений, где он присутствует. Например, для обучения модели на маскоте, нужно собрать датасет, включающий изображения, где он присутствует и описать их соответствующим образом. Например: 1.txt: photo of babrich, jacket, jeans; 2.txt: photo of babrich, shorts, t-shirt. При обучении модели нужно использовать метки, такие как babrich, для идентификации этого персонажа в текстовых описаниях.

Если тренируете модель на авторских стилях, так же, как и с персонажем, нужно собрать датасет, состоящий из изображений, созданных данным художником, и описать их стиль в текстовых подсказках. Например: ArtistName, mountains, night, moon, snowy peaks, stars и т. д. Вызов стиля художника через ArtistName.



1.png



1.txt



2.png



2.txt



3.png



3.txt



4.png



4.txt



5.png



5.txt



6.png



6.txt

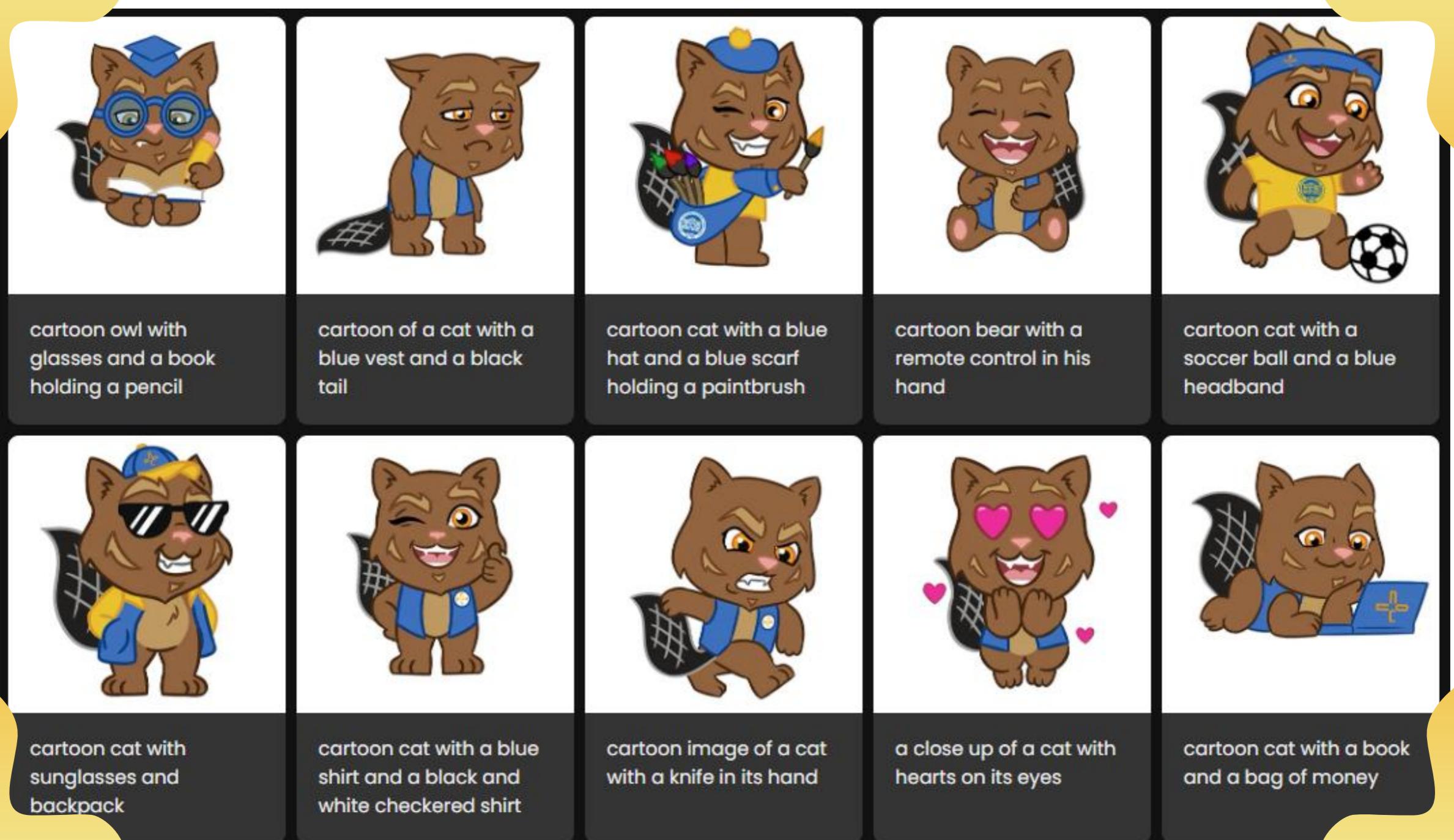


7.png



7.txt

После того как датасет с нужными изображениями готов, перемещаем их в область справа. Для примера использовался набор стикеров ППОС ИГУ из Telegram.

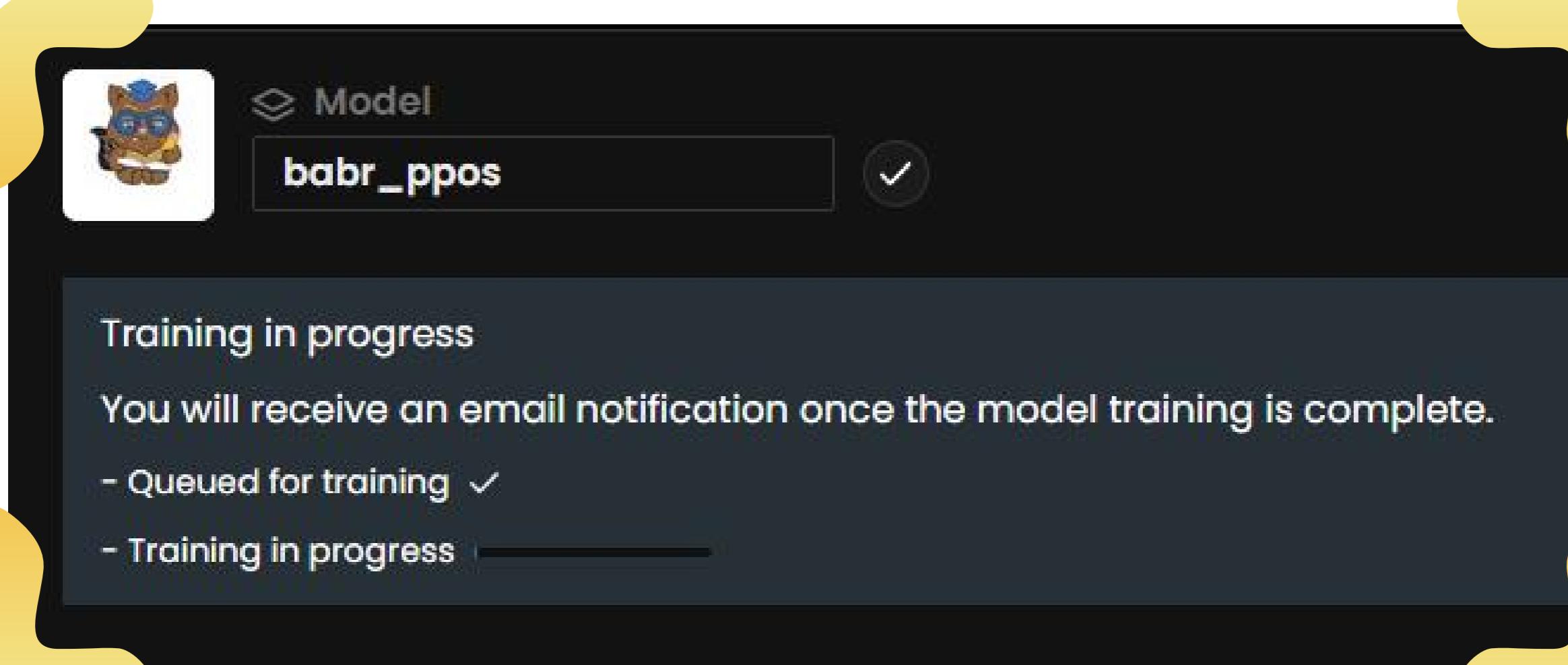


Сервис автоматически подбирает описание к загруженным изображениям, но оно не всегда верное, поэтому лучше перепроверить, добавить что-то в описание.

ТРЕНИРОВКА

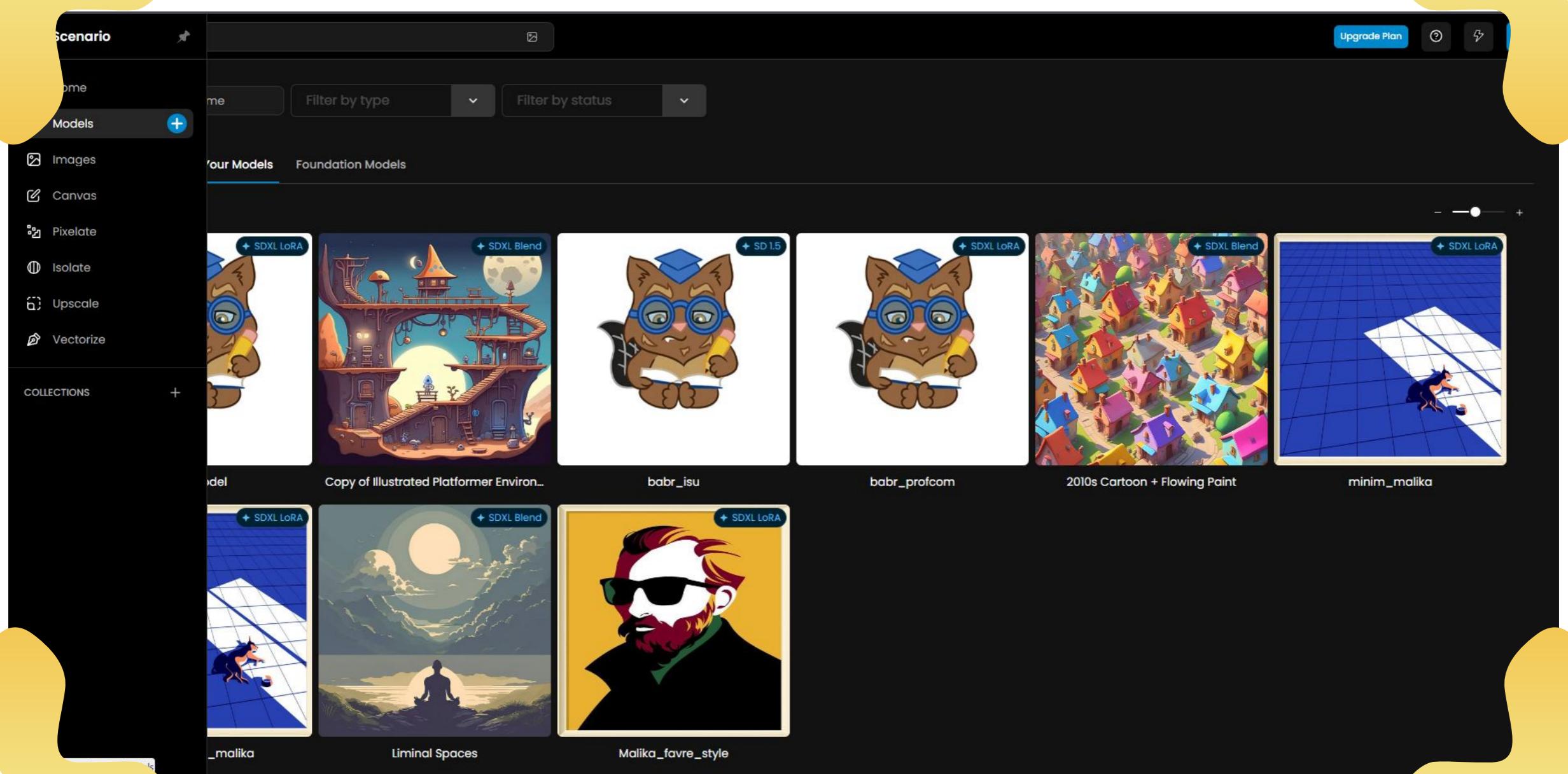
LORA

После того как все настроили и подготовили датасет, нужно вписать название для модели и нажать на кнопку Start Training.

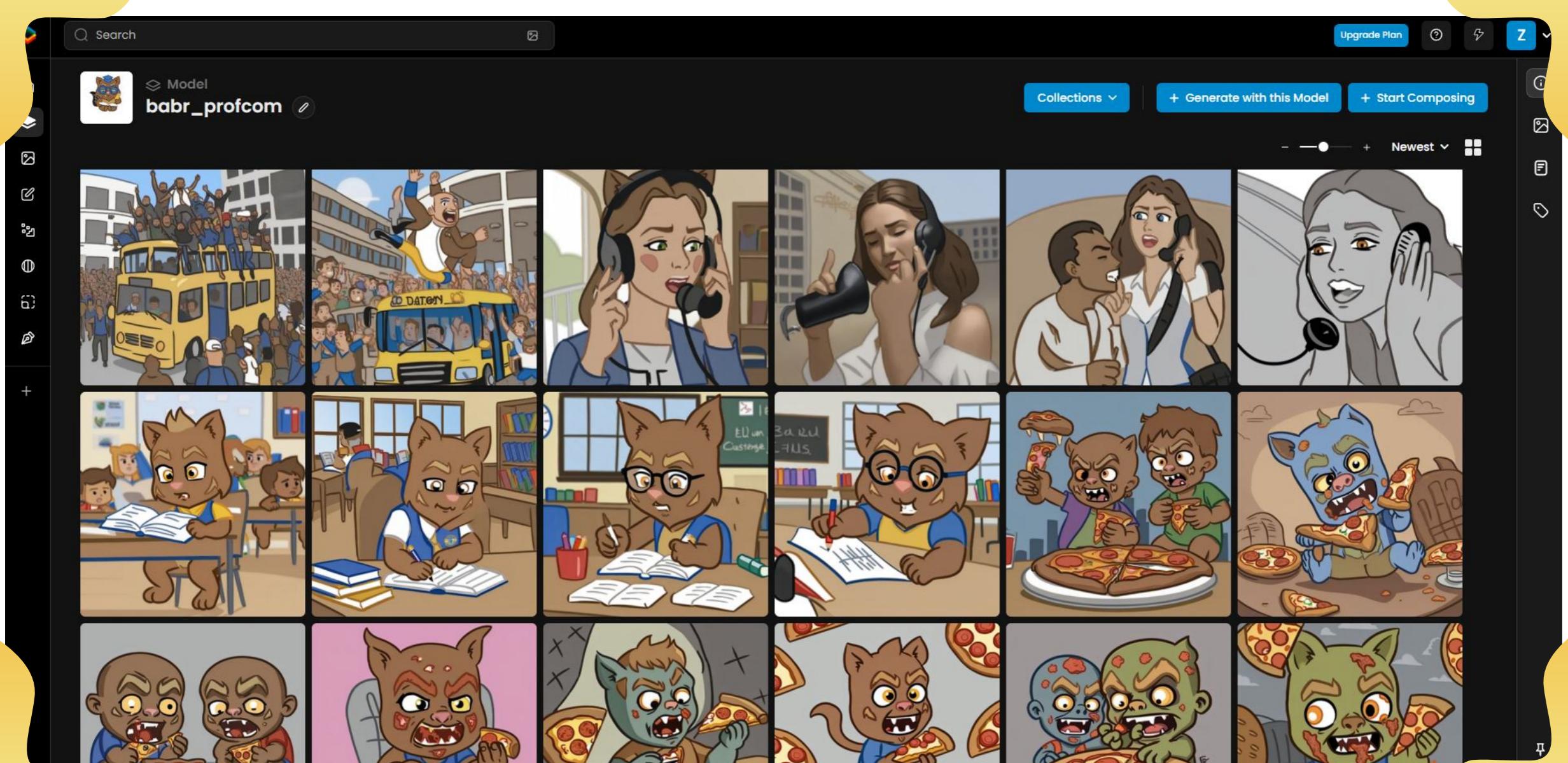


По окончании тренировки на почту, указанную при регистрации, придет письмо о готовности модели. Если не закрывать страницу, то можно наблюдать за процессом тренировки в реальном времени.

После окончания тренировки модель можно найти на вкладке models.

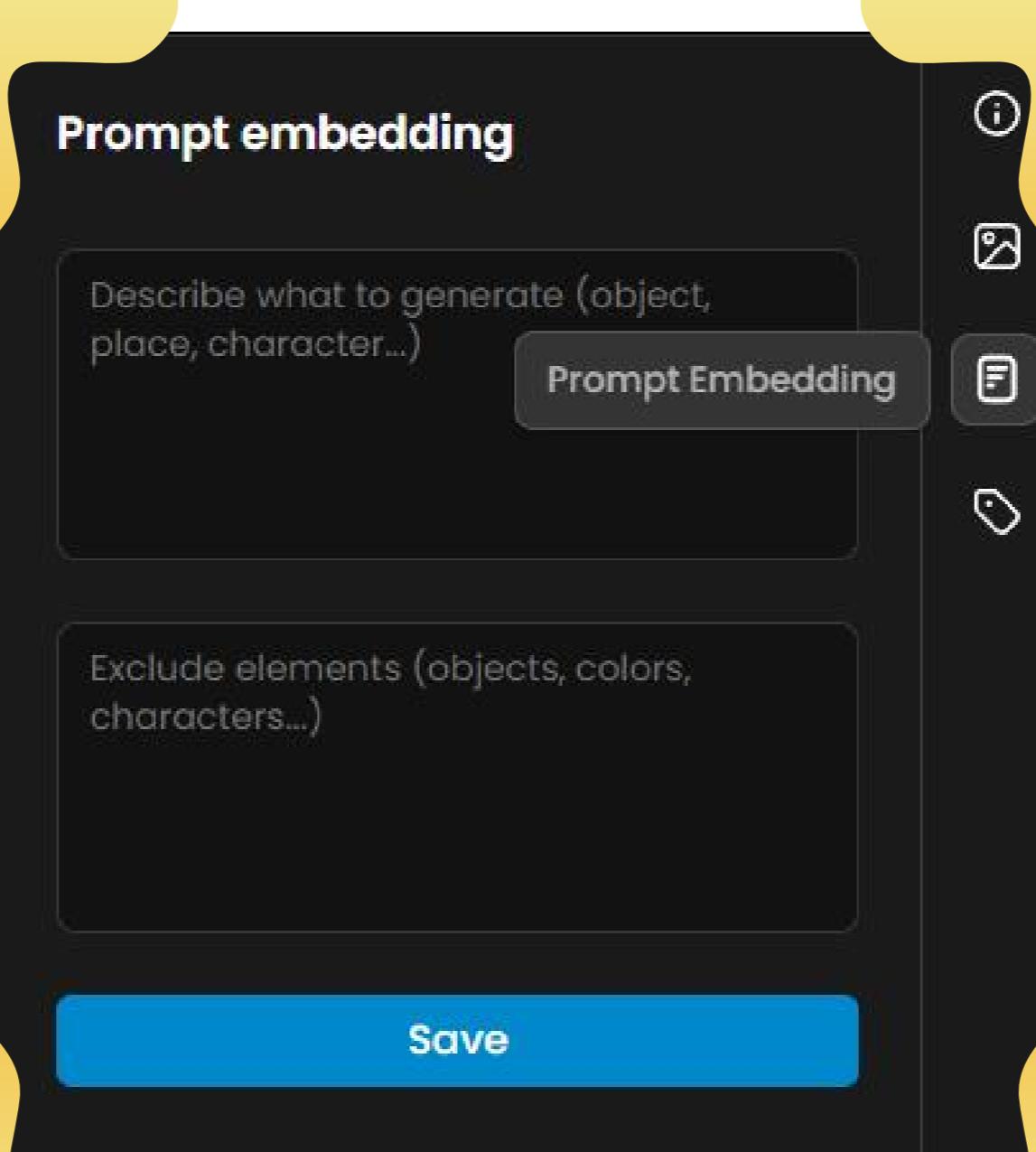


При нажатии на модель, открывается вкладка Overview, в которой можно закрепить лучшие примеры изображений. Также есть вкладка images, в которой будут храниться сгенерированные изображения и вкладка Details.



PROMPT EMBEDDING

Во вкладке *Details* у модели есть раздел с информацией о модели, а также раздел "Prompt Embedding", где можно написать подсказку для генерации изображения. Это позволяет быстро сохранять пользовательские выражения и делиться ими с командой для улучшения работы. Здесь определяются ключевые слова или "фишки", чтобы указать модели, какие изображения и как их создавать. Это помогает поддерживать единый стиль и тематику без необходимости переобучать модель.

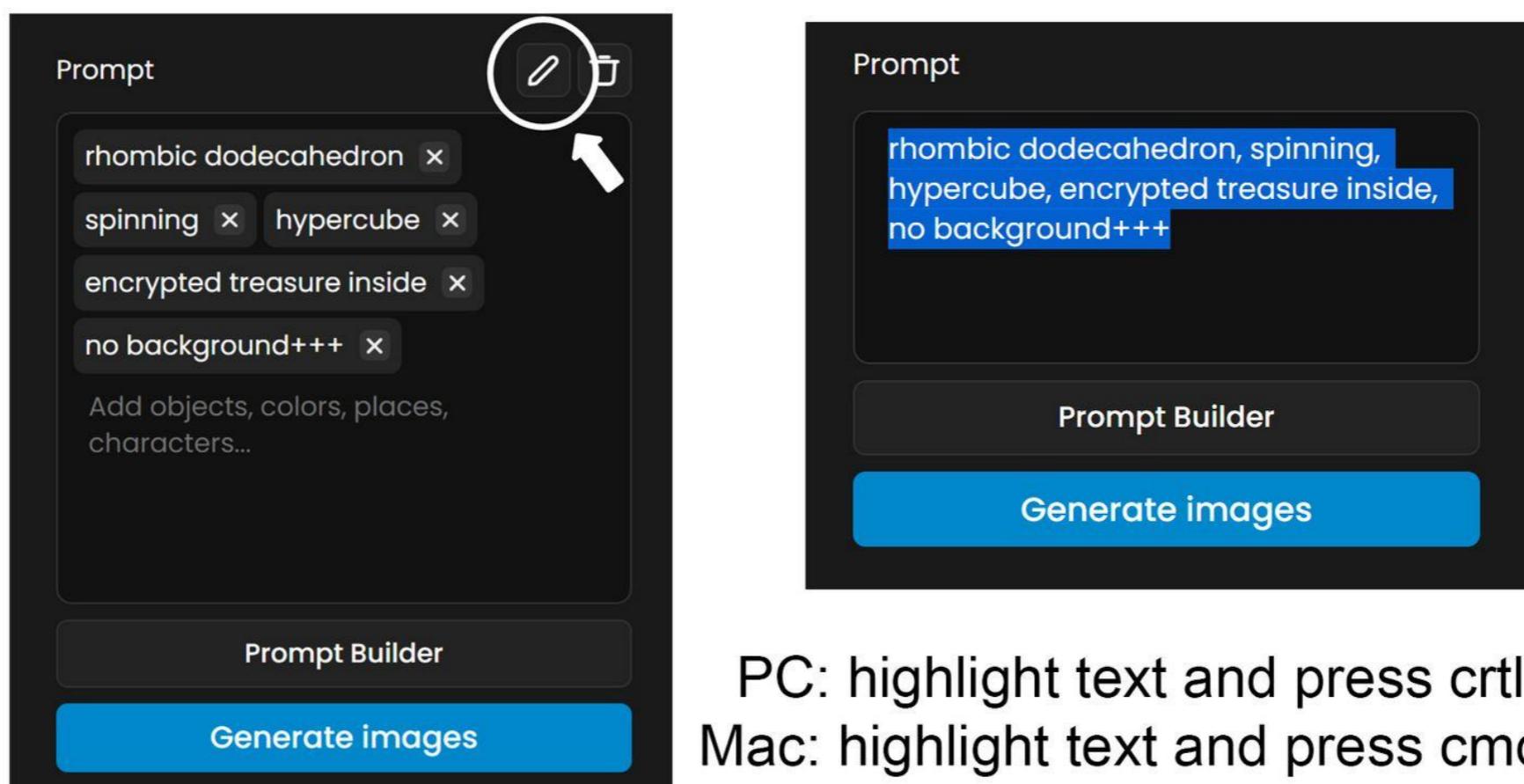


ПРИМЕР

1

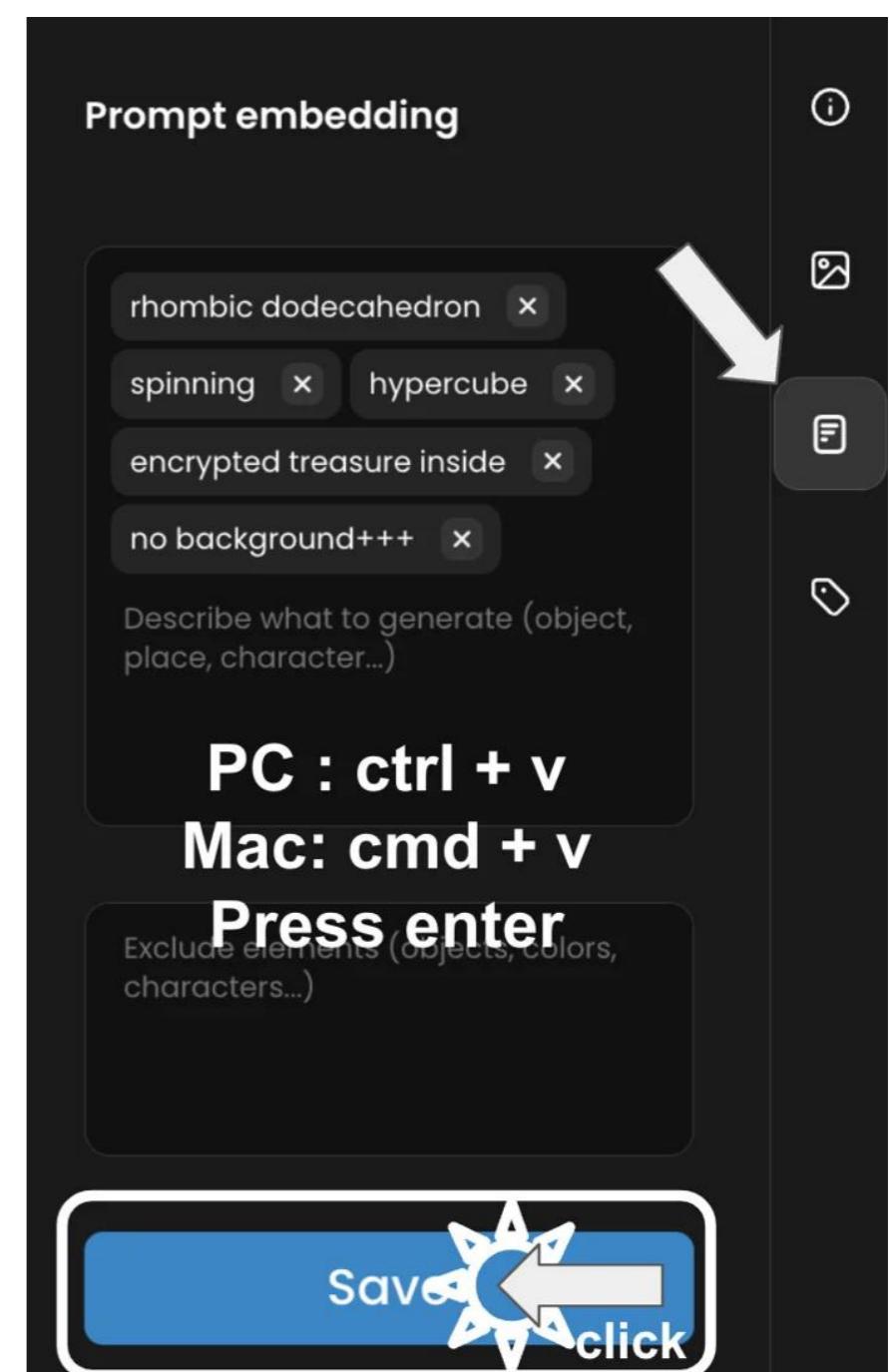
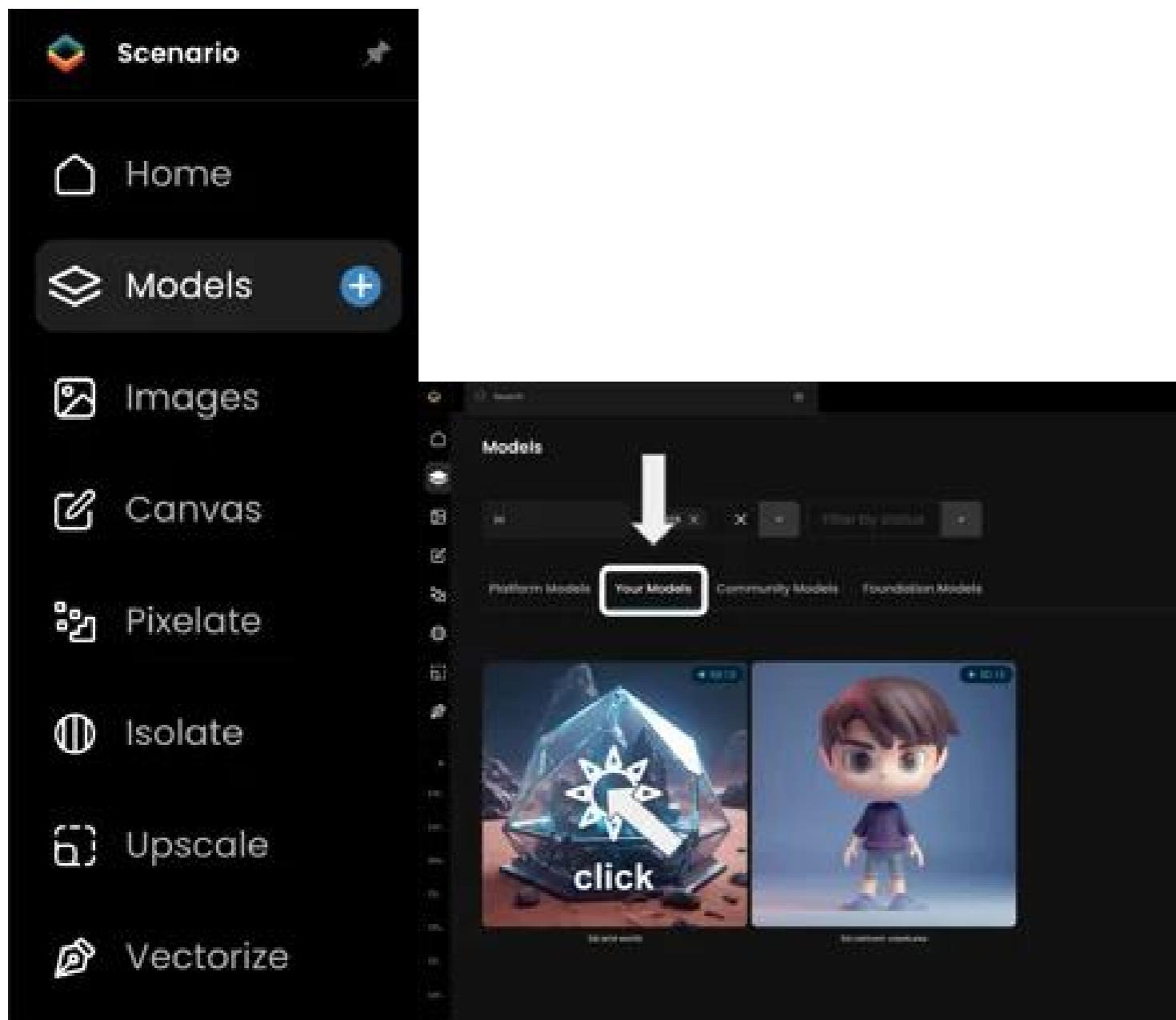


2



PC: highlight text and press **crtl + c**
Mac: highlight text and press **cmd + c**

3



Теперь можно генерировать изображения с помощью обученной модели, загрузив предварительно свои подсказки для контроля над более точным процессом генерации изображения.

Prompt

Gold



Garnet



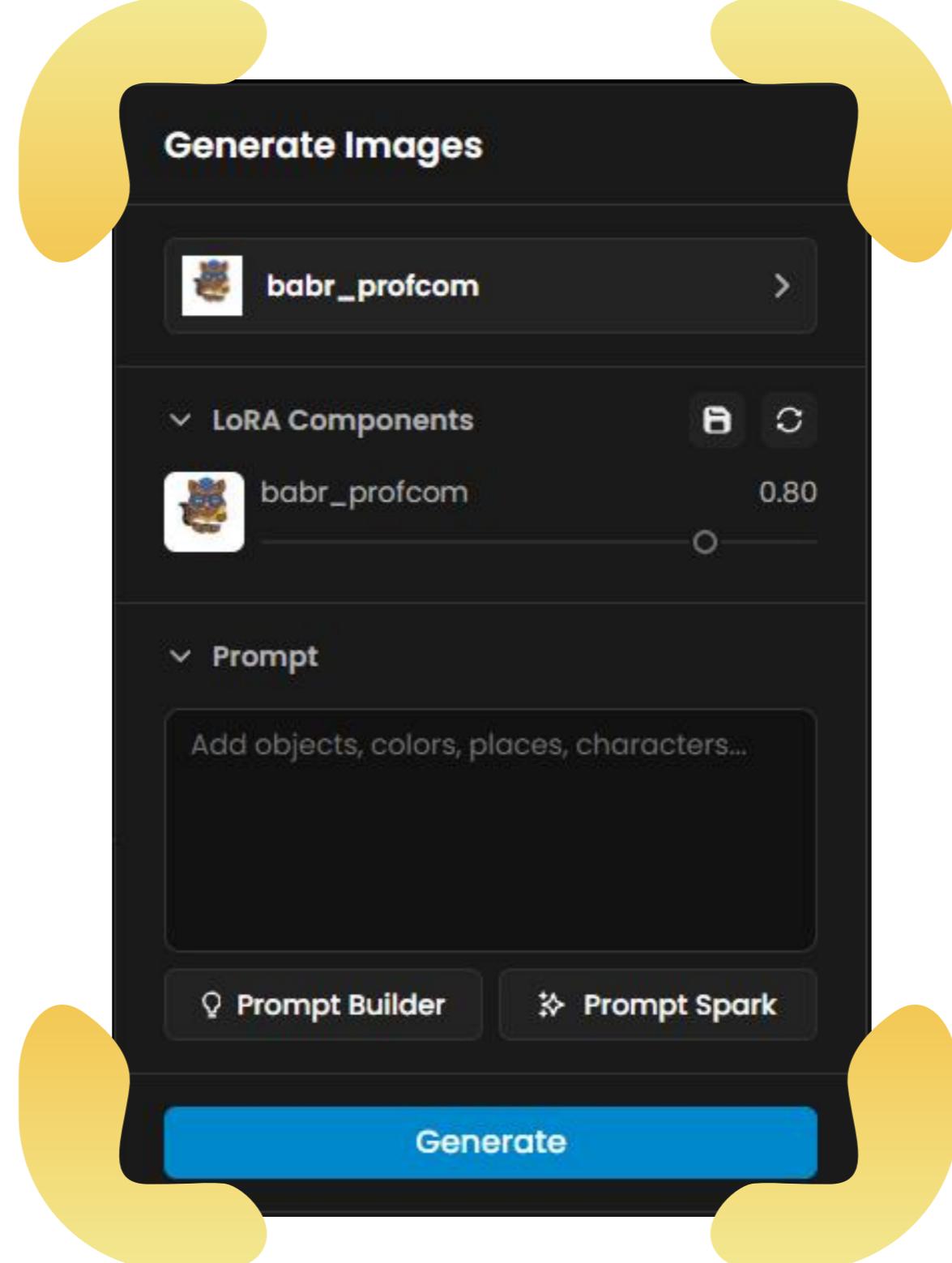
Amethyst



Pearl



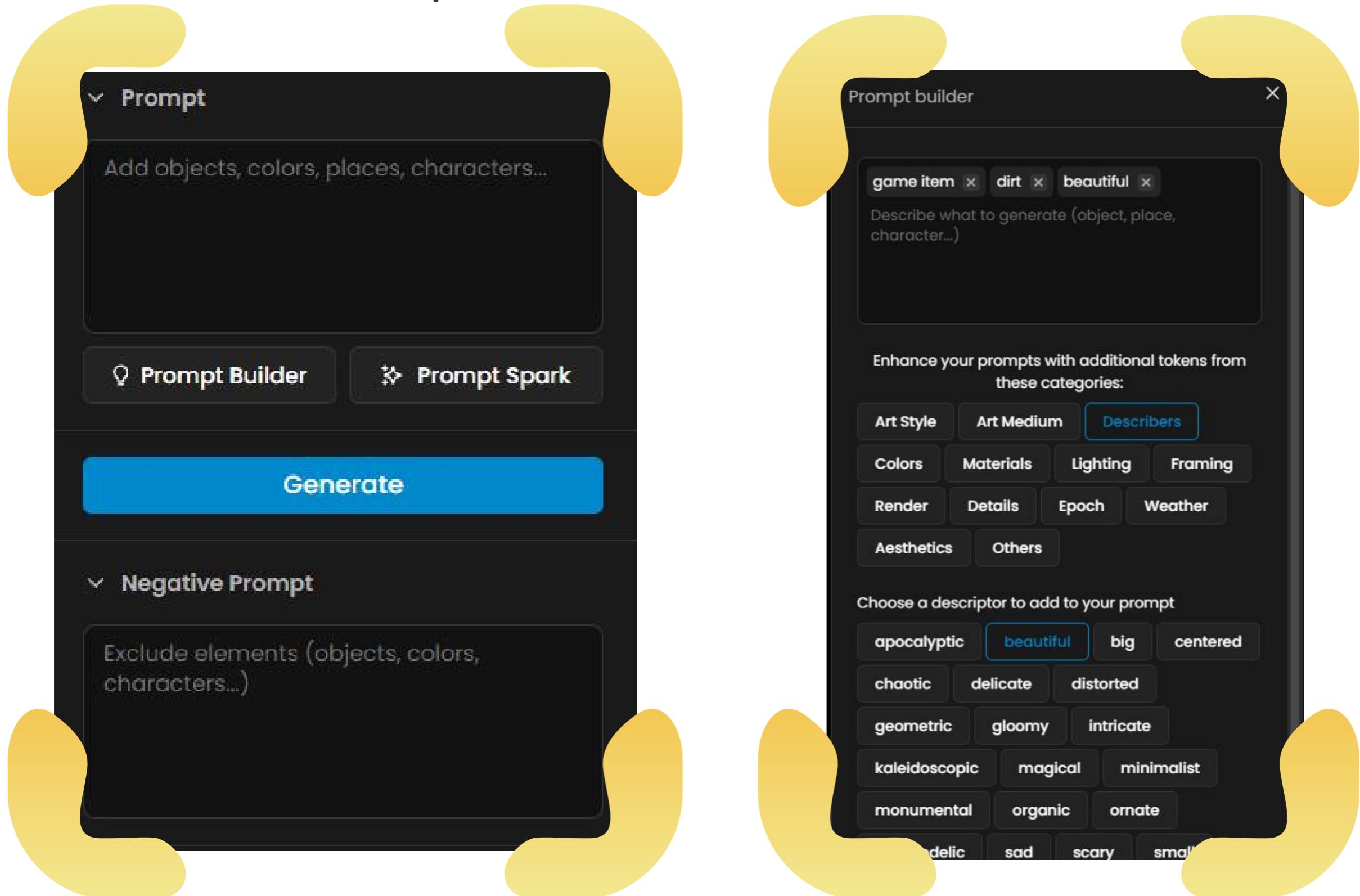
Нажимаем на Generate with this model. Открывается страница с генерацией изображений. В левой части экрана расположены настройки генерации, справа располагаются сгенерированные изображения. Приведем описание **настроек**:



Ползунок рядом с названием модели означает, насколько сильно будет применяться стиль модели к генерации. Чем меньше, тем слабее будет влиять LoRA на изображения.

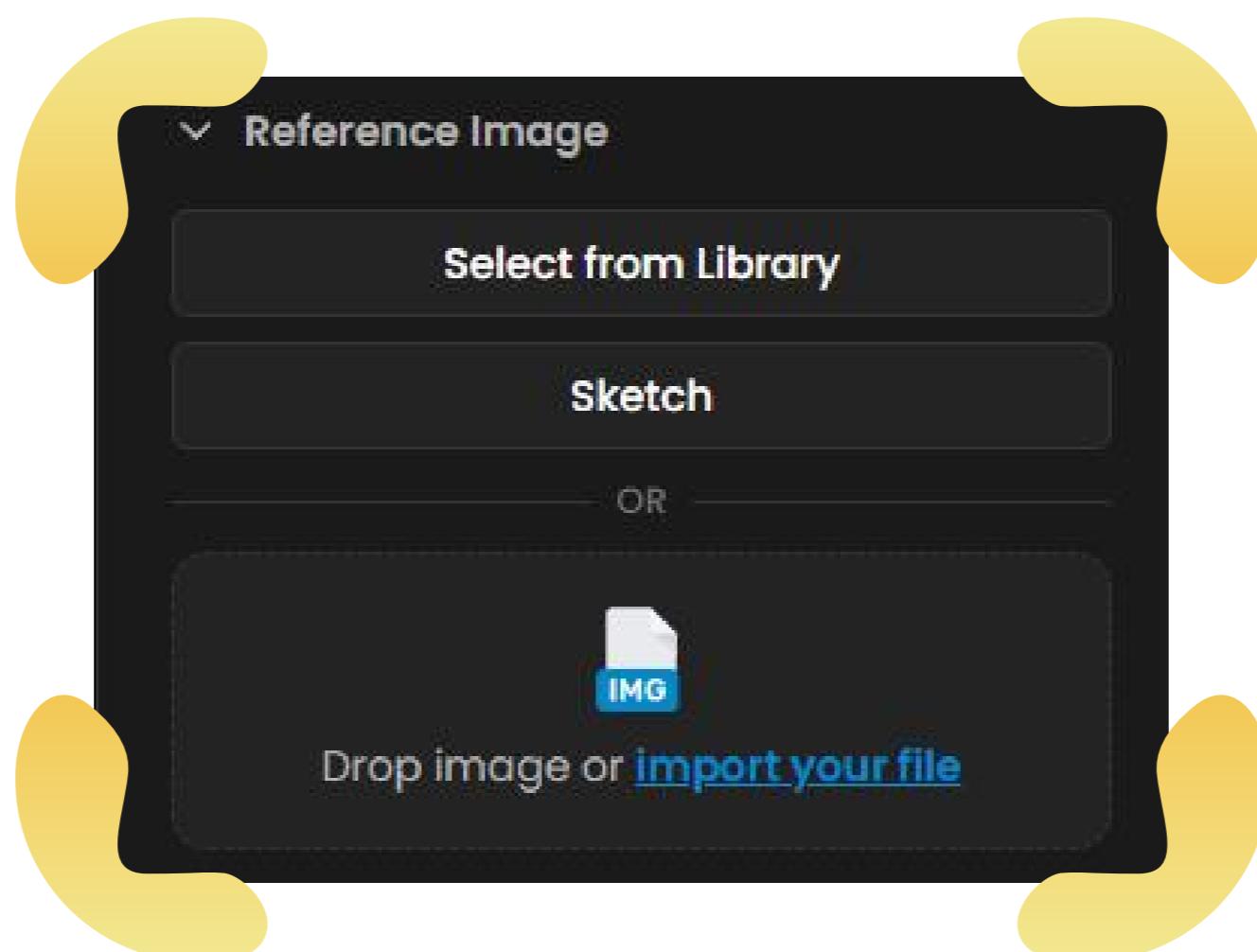
Promt: это текстовое описание того, что нужно сгенерировать, это может быть как что-то очень примитивное, например, «слон в джунглях», так и профессиональные промты. Обширная тема, можно изучить в интернете как правильно это делать. Самое главное что нужно знать в нашем случае, это то, что нужно вписывать триггер слово, которое выбрали в описании изображений. Для примера это babrich. Оно обязательно должно быть написано, иначе стиль LoRA не применится.

При затруднениях с генерацией запросом на помощь может прийти кнопка Promt Spark, и сервис сгенерирует запрос сам, а кнопка Promt Builder поможет в составлении запроса.

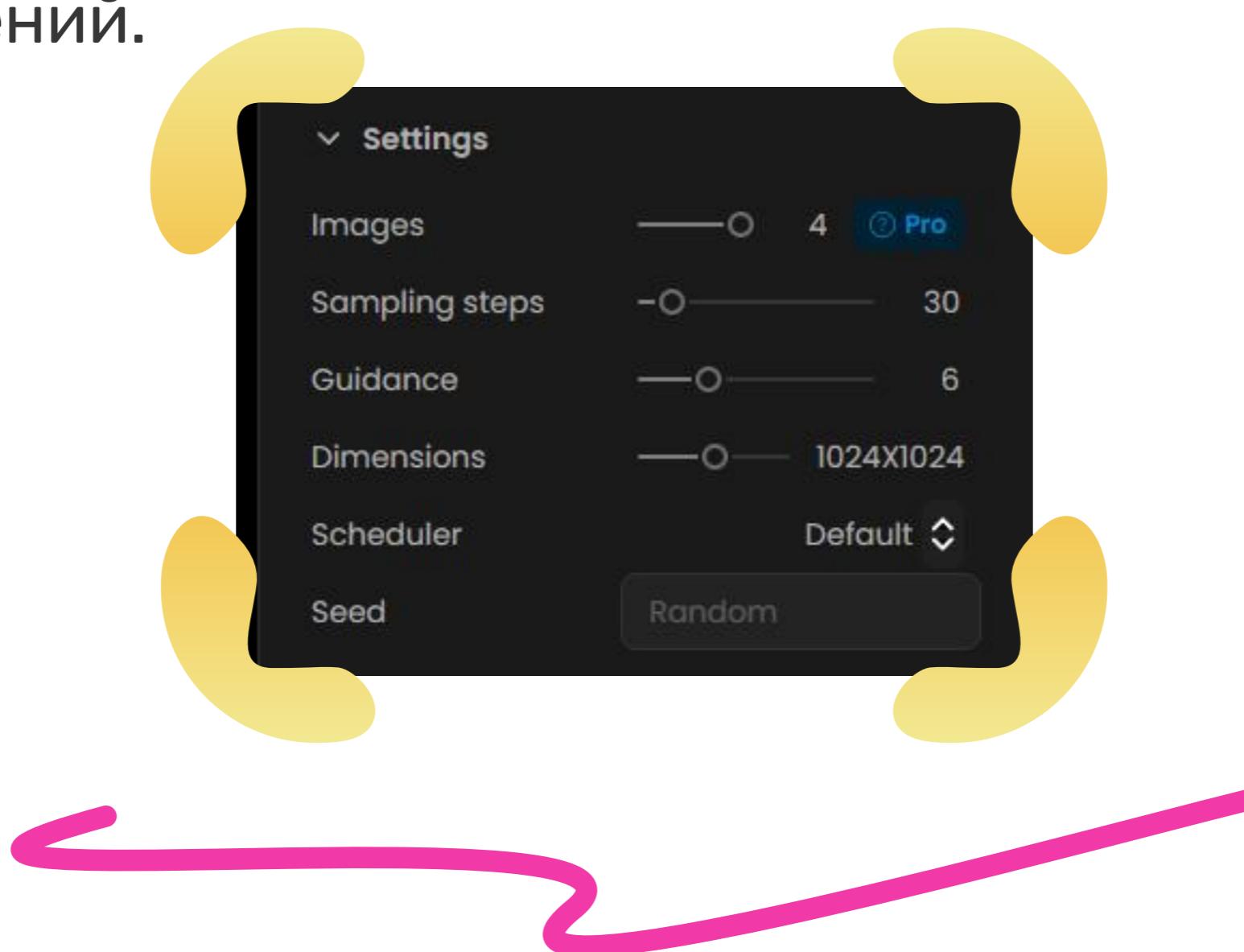


Negative Promt: негативный промпт в Stable Diffusion - это ключевые слова или фразы, которые указываются, чтобы предотвратить генерацию определенных элементов, стилей или объектов в создаваемых изображениях. Фактические, здесь описываются элементы, которые не должны быть включены в итоговое изображение, помогая настроить результаты генерации в соответствии с представлением пользователя. Как правильно составлять такие запросы можно посмотреть в интернете.

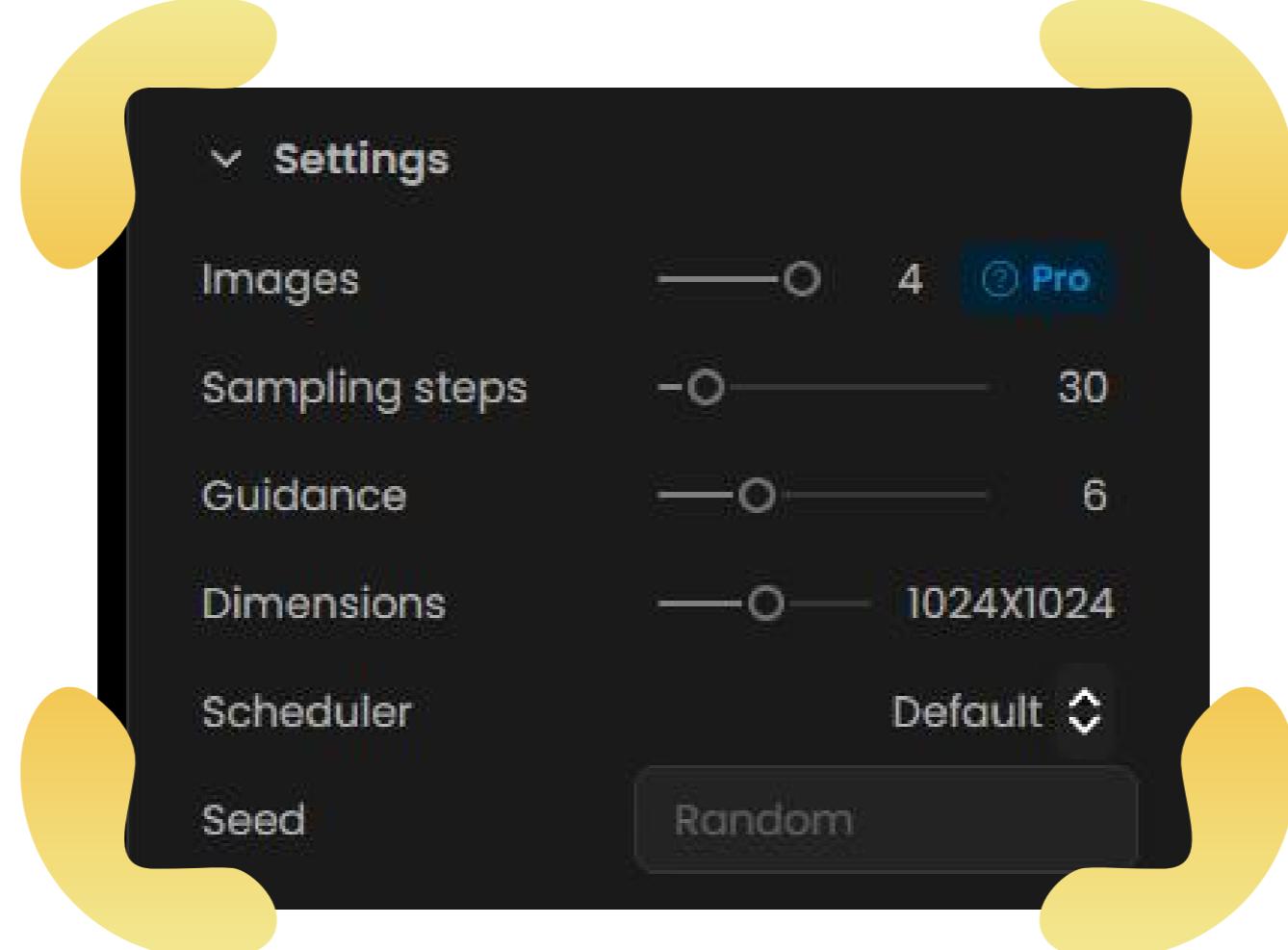
Reference Image: это функция, которая позволяет использовать изображение в качестве ориентира для стиля и композиции. Можно выбрать, насколько близко или различно результат будет похож на исходное изображение. Данный инструмент способен вносить микро-коррекции, а также полностью трансформировать исходное изображение.



Images: выбираем сколько нужно сгенерировать изображений.



Sampling steps: количество шагов по созданию изображения. Чем их больше, тем дольше нейросеть будет над ним работать. Не стоит устанавливать это свойство больше 30–40, иначе нейросеть будет вести себя как художник, который бесконечно мажет одно и то же пятно.

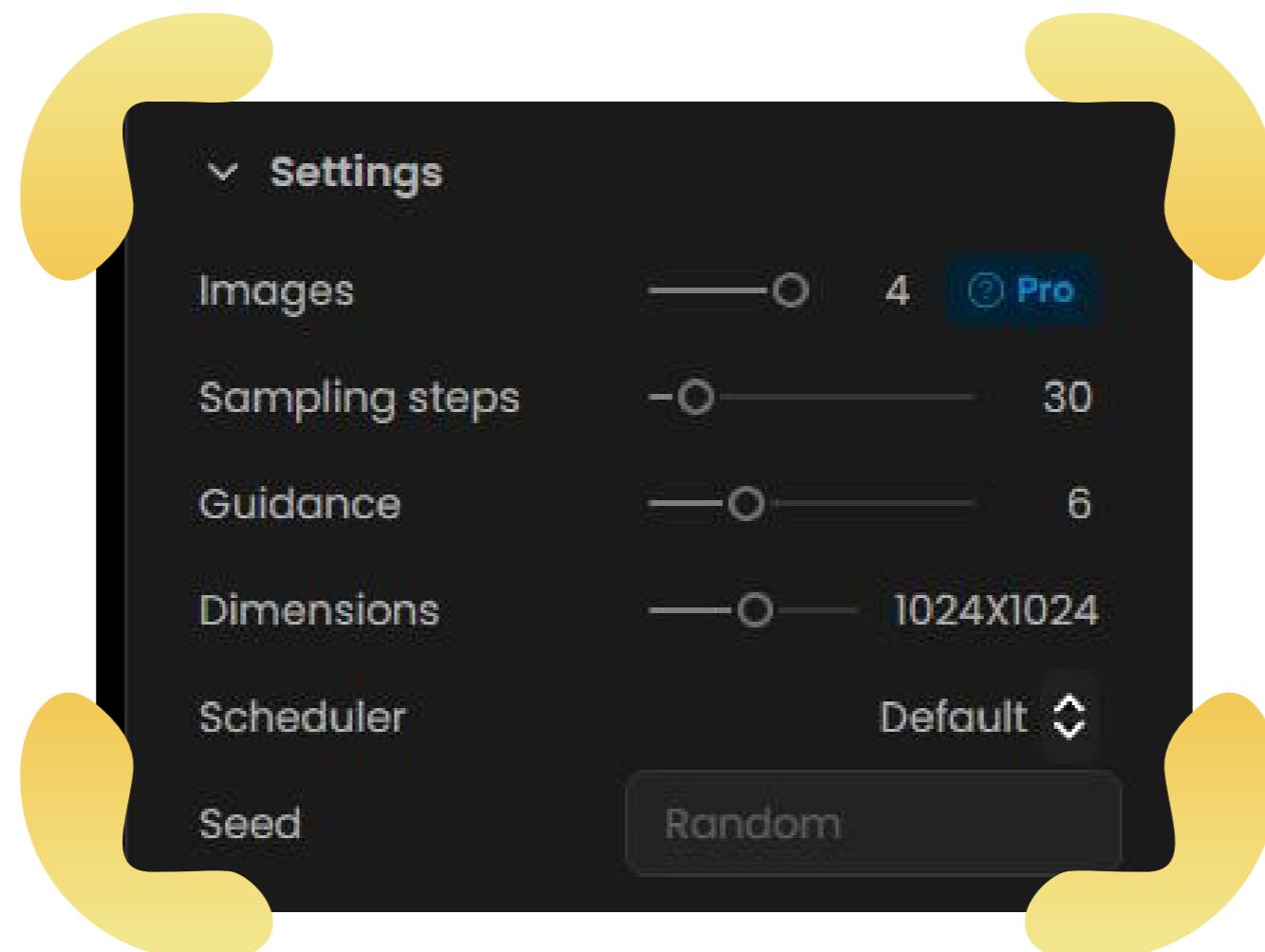


Guidance: определяет, насколько сильно процесс генерации ориентирован на подсказку. Значение 0 эквивалентно отсутствию подсказки, и изображение полностью не связано с подсказкой (но оно все равно может быть интересным). Увеличение значения повышает степень сходства изображения с подсказкой, но приводит к перенасыщению и резкости изображения.

Dimensions: разрешение сгенерированного изображения.

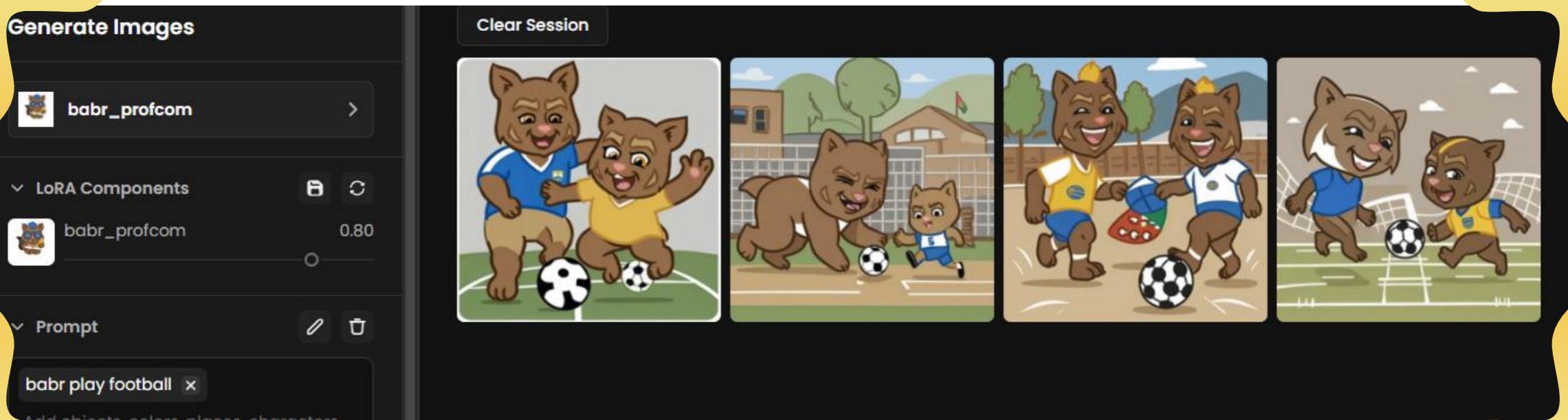
Scheduler: это метод, который вы выбираете для денойзинга шума, чтобы получить абсолютно новое изображение. Обычно рекомендуется использовать стандартный планировщик для большинства случаев. Настройка планировщика предназначена для опытных пользователей, так как это очень глубокая тема.

Seed: У каждого процесса есть начало. В алгоритмах графических нейросетей начало состоит из случайного числа, называемого "зерном". Это, можно сказать, похоже на белый шум на телевизоре: множество светлых и тёмных точек, случайно покрывающих экран.

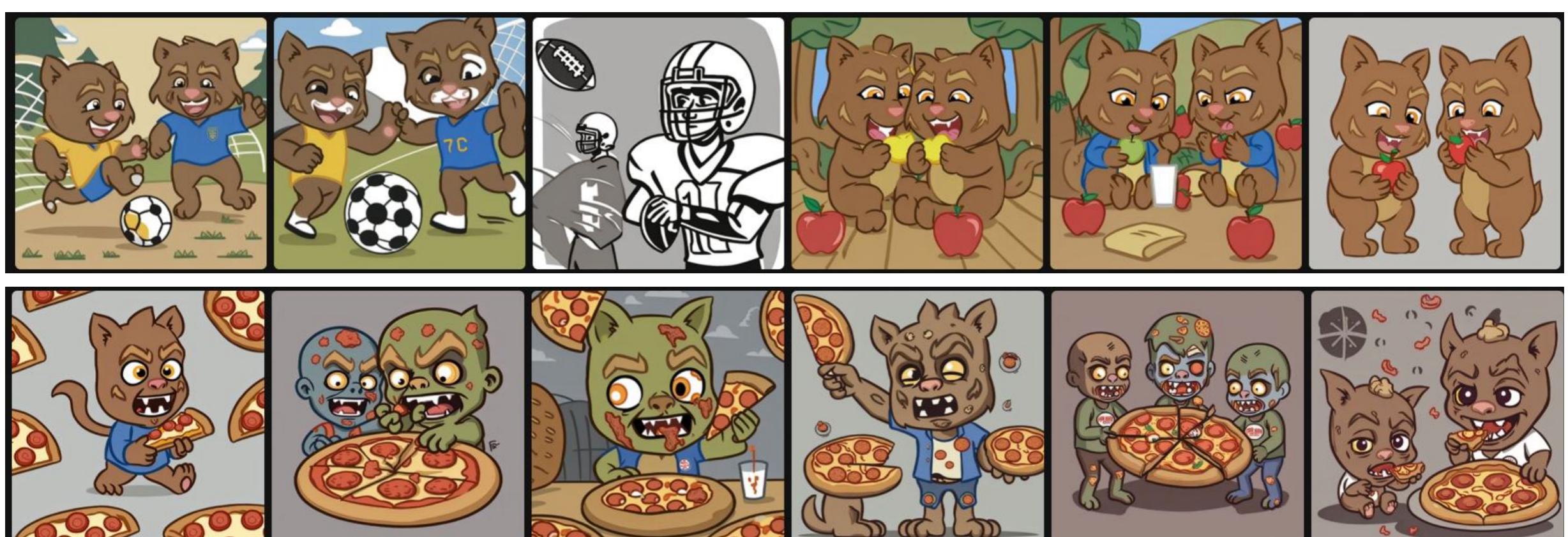


Точно также начинается генерация изображения, и "зерно" описывает начальную комбинацию точек. По умолчанию каждый раз "зерно" разное: оно генерируется случайно. Но можно ввести его вручную или скопировать из данных какого-то конкретного изображения, созданного ранее. Зачем? Если использовать одно и то же "зерно" с одинаковым запросом (ничего не меняя), будет получаться одно и то же изображение снова и снова. Таким образом можно получить несколько вариаций одного и того же.

РЕЗУЛЬТАТ



Вот результаты сгенерированных изображений на примере датасета со стикерами ППОС ИГУ. С учетом того, что все настройки выставлены стандартные, датасет невелик и написаны небольшие промты для генерации, это вполне приемлемый результат, который не требует много времени.





ТРЕНИРОВКА

LORA

НА СВОЁМ ПК

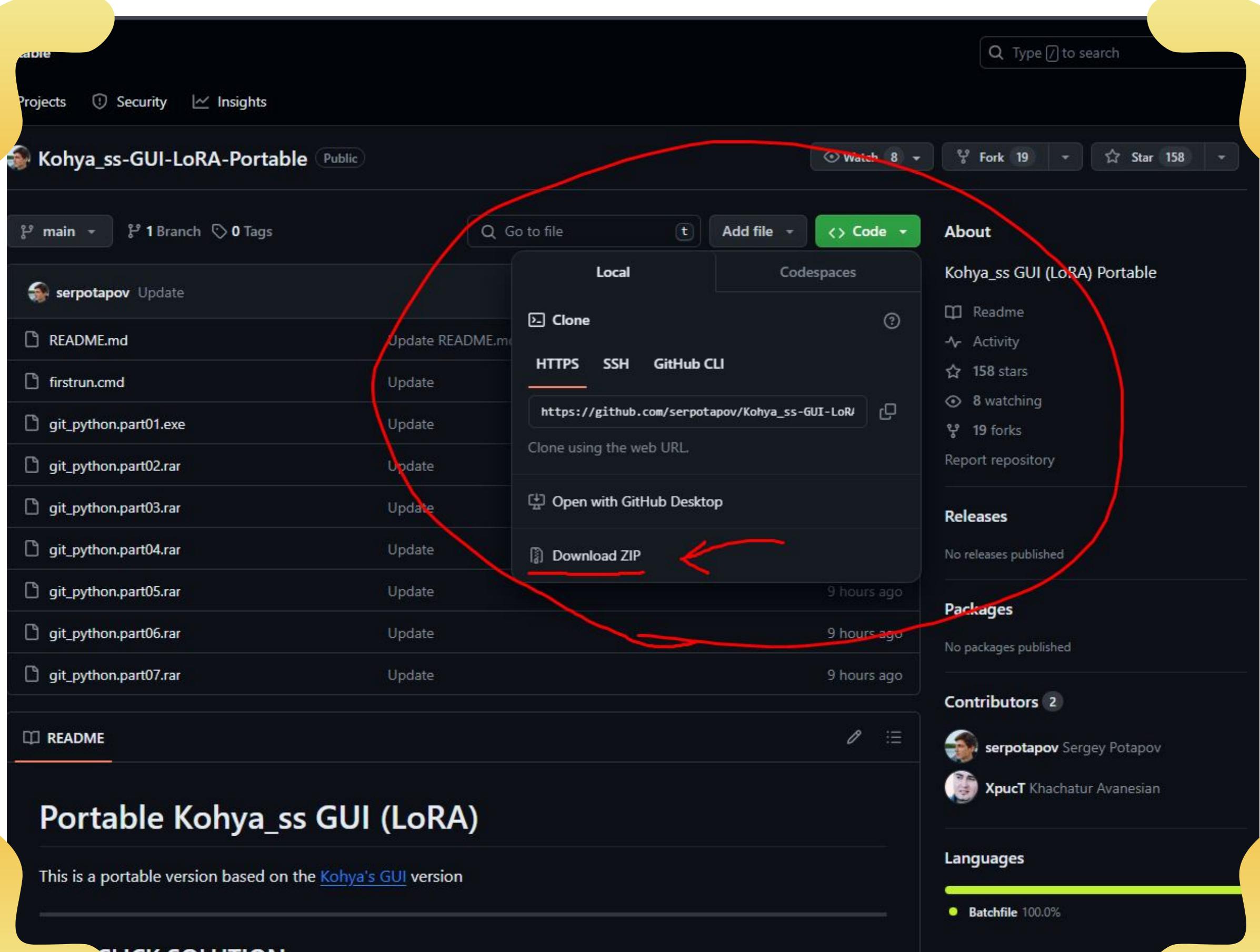
Не подходит для новичков

УСТАНОВКА

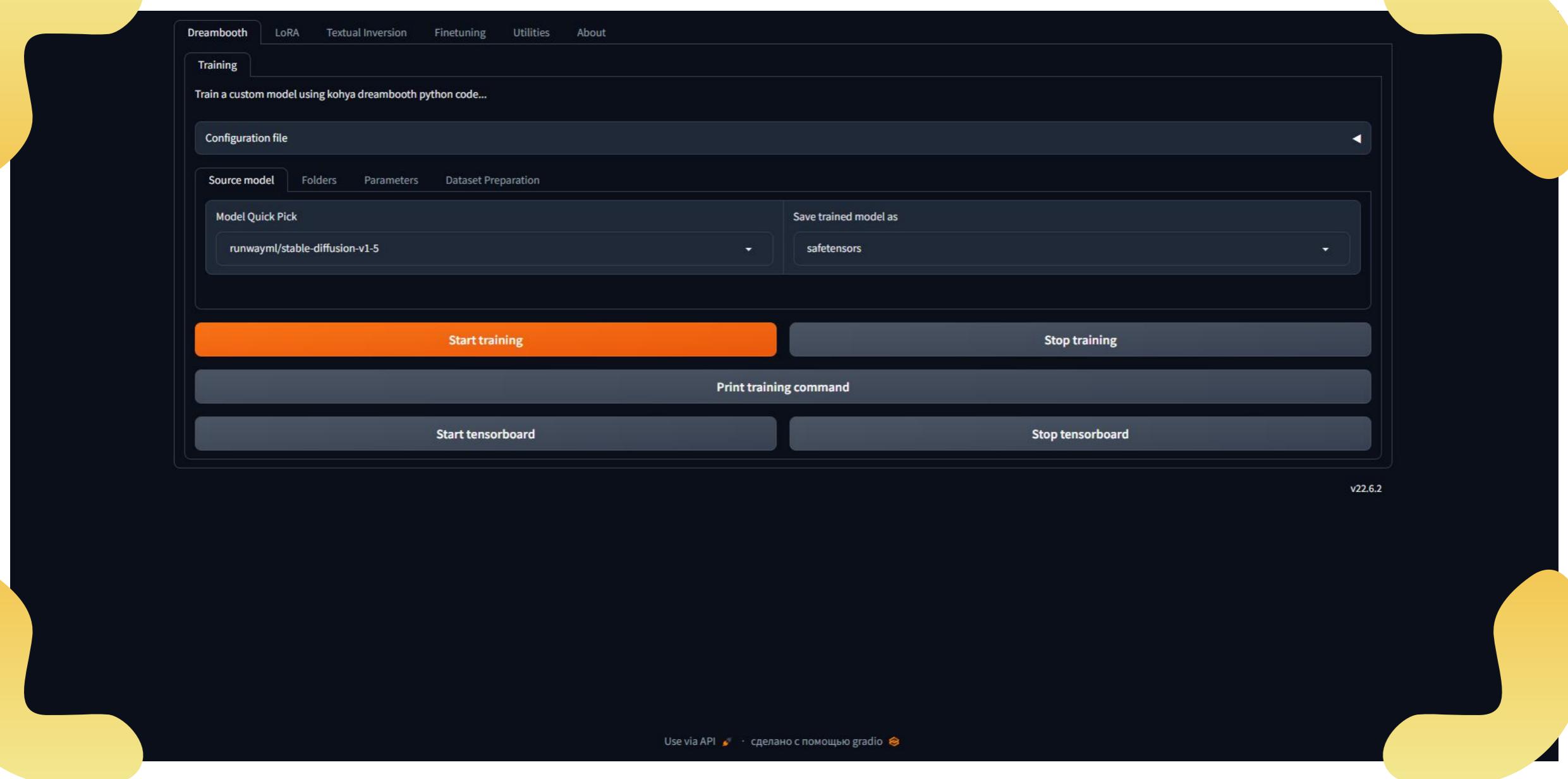
ПО

Для начала нужно сказать, что этот способ не подойдет неопытным пользователям, а также для слабых ПК. Чтобы хоть что-то запустилось, понадобится минимум 6 GB видеопамяти у Вашей видеокарты.

Сперва, для тренировки LoRA, нужно установить программу Portable Kohya_ss GUI с гит репозитория ([ссылка 2 на с.43](#)).

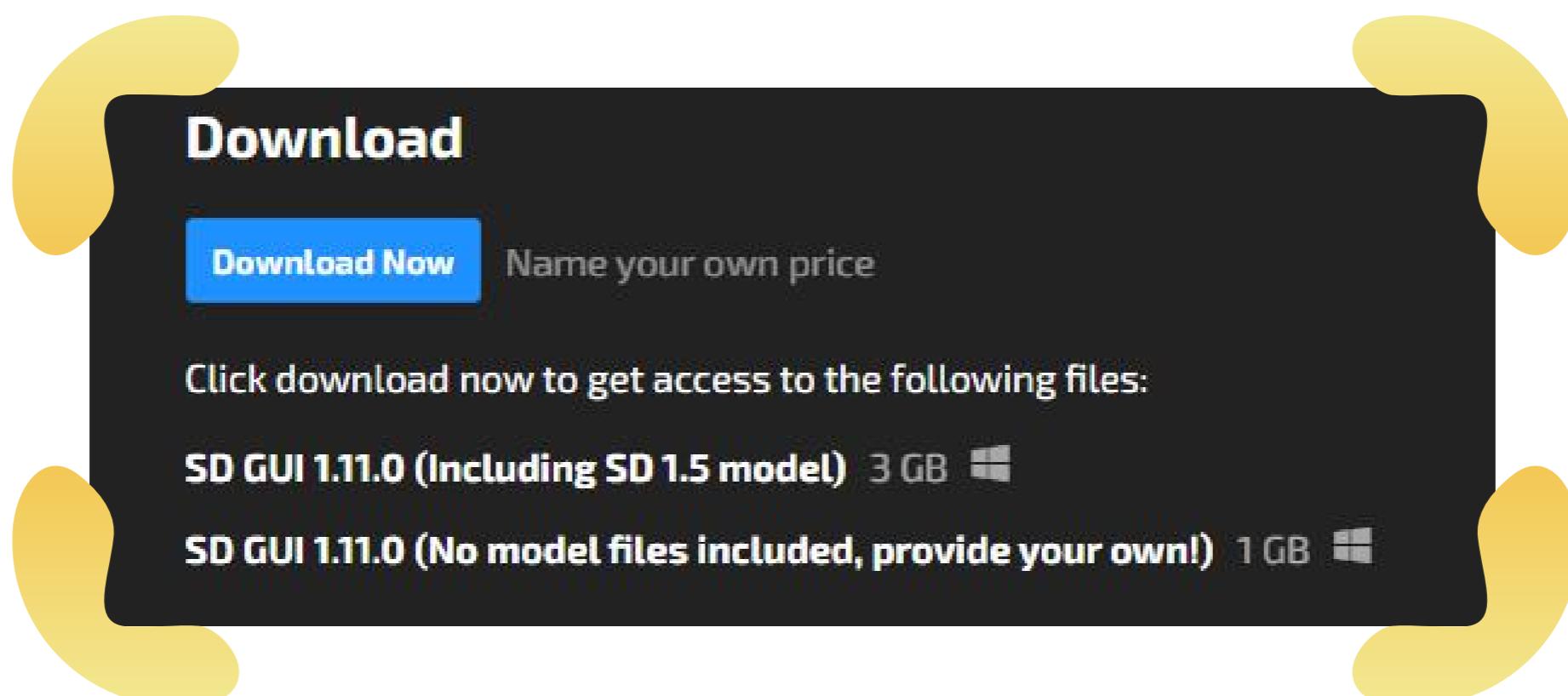


После скачивания и распаковки архива запускаем файл **firstrun** и ждем пока все скачается и установится.



По окончании установки, откроется интерфейс программы в браузере. Пока что ничего не нажимаем, переходим к установке ПО для генерации изображений. Для этого нужно зайти по ссылке **3 на с.43** и скачать программу NMKD GUI.

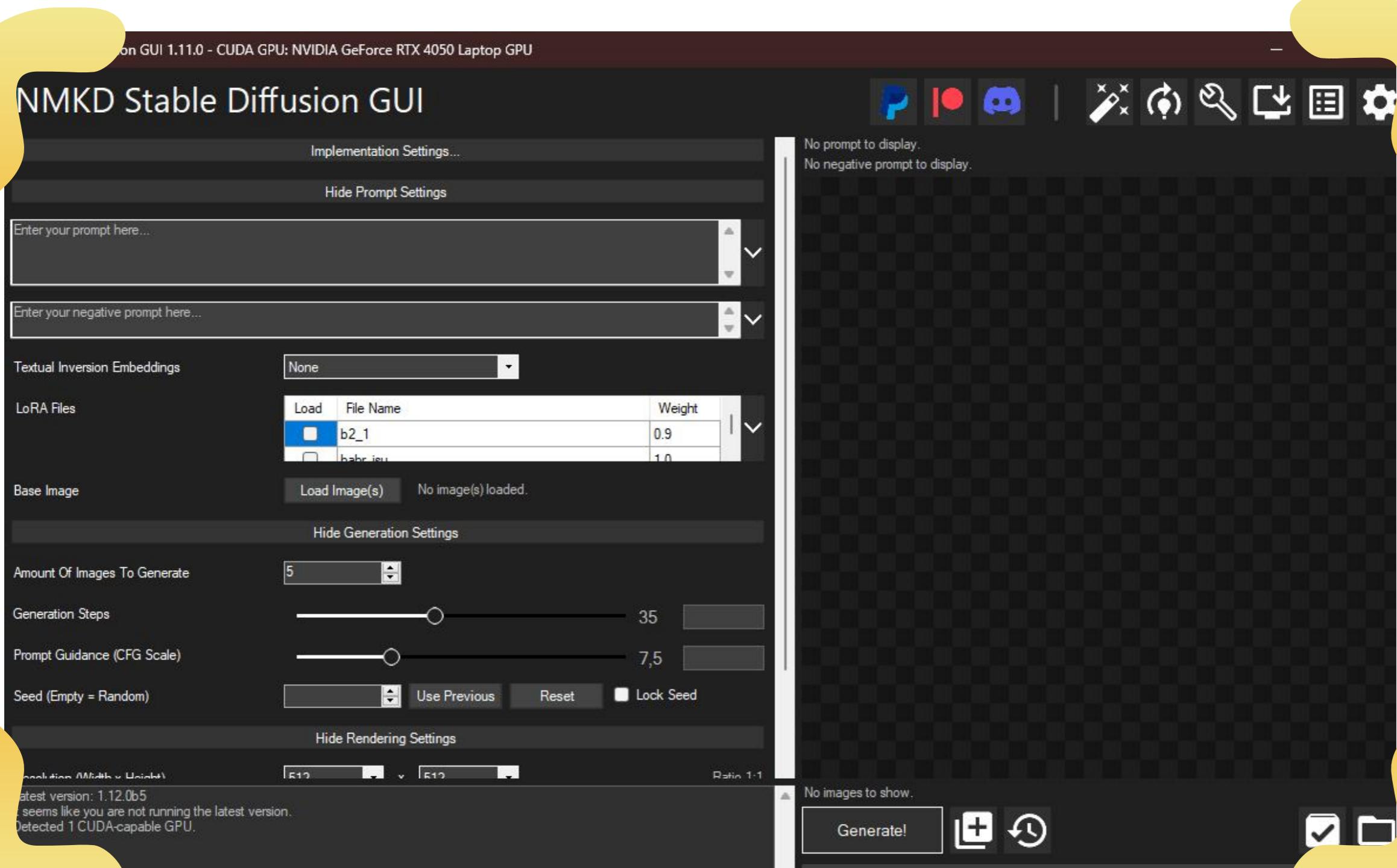
Чтобы генерировать изображения через программу, потребуется модель Stable Diffusion. В следующем рассматриваемом примере это SD 2.1, но можно скачать любую (с учетом того, что чем лучше модель, тем больше ресурсов потребуется для тренировки LoRA).



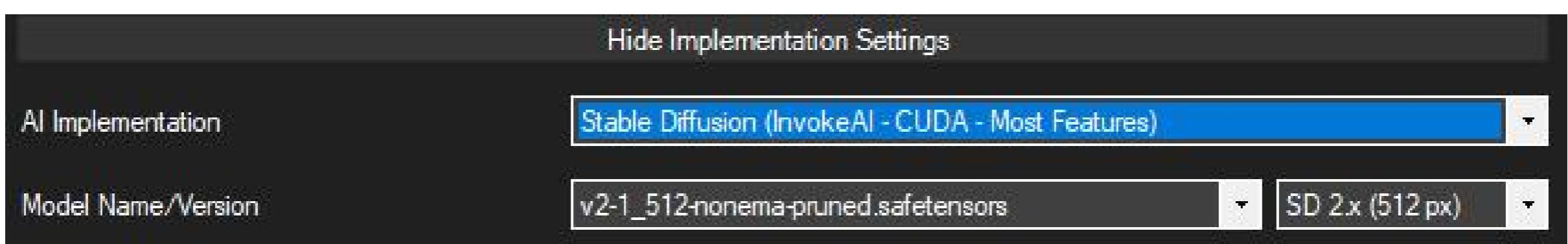
Если никакой модели нет на примете, можете скачать NKMD, включающую в себя SD 1.5.
Скачиваем, распаковываем архив.

ОБЗОР ИНТЕРФЕЙСА

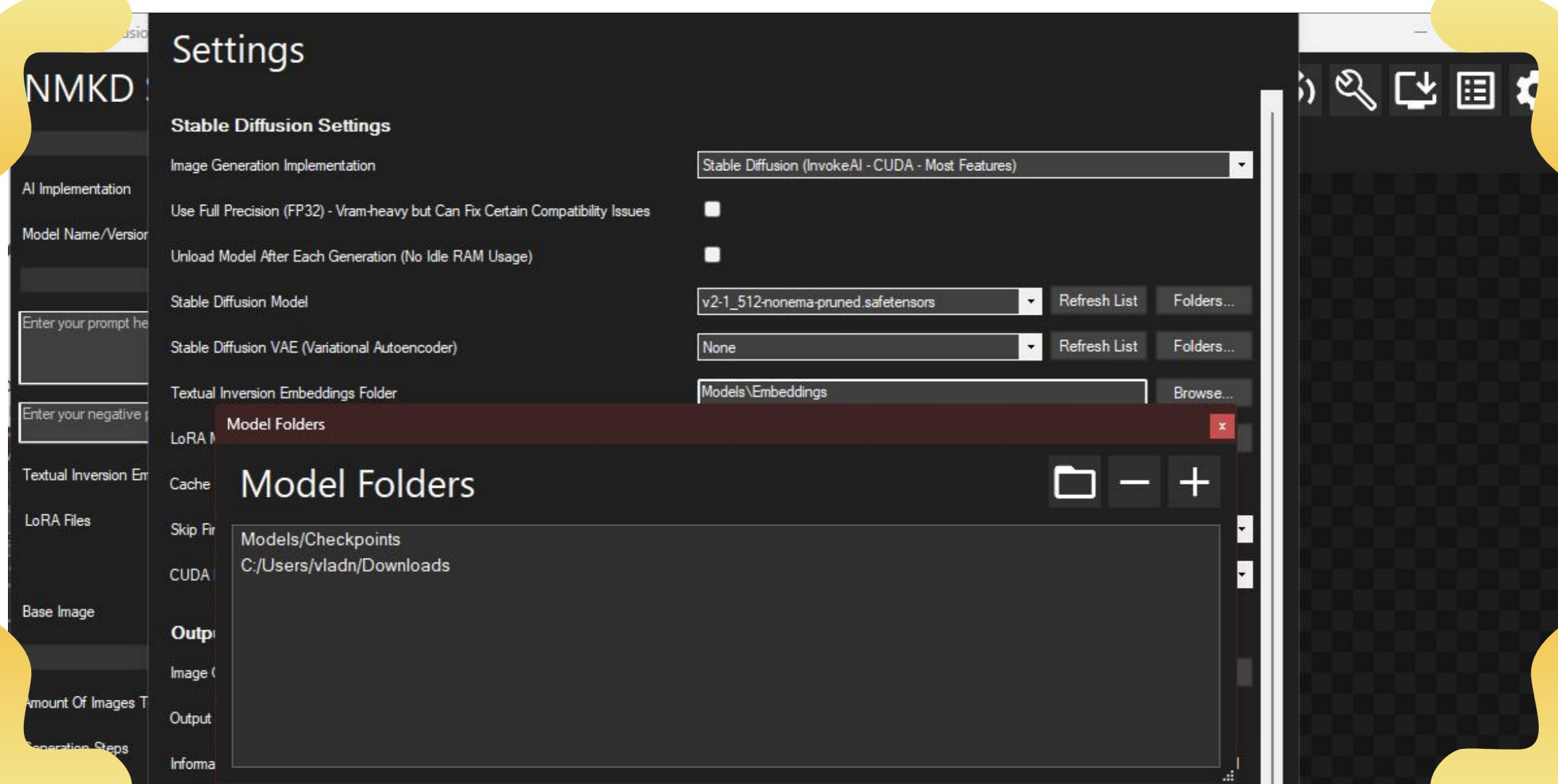
После установки нужного ПО (программного обеспечения), нужно ознакомиться с тем, что оно предлагает. Начнем с NKMD.



Справа видим область, в которой будут появляться сгенерированные изображения, а слева **настройки** генерации. Начнем с **Implementation settings**: здесь нужно выбрать какая модель будет использоваться для генерации в пункте **Model Name**, а также нужную версию справа.



Чтобы добавить скачанную модель, нужно переместить её в папку NKMD/models/checkpoints либо зайти в настройки и указать папку, из которой программа будет брать модели.



AI Implementation оставляем как есть для NVIDIA, но для видеокарт AMD ставим ONNX Direct ML.

Далее идут **Promt Settings**, которые почти ничем не отличаются от тех, что расписаны на [стр. 22](#).

Появились два новых пункта.

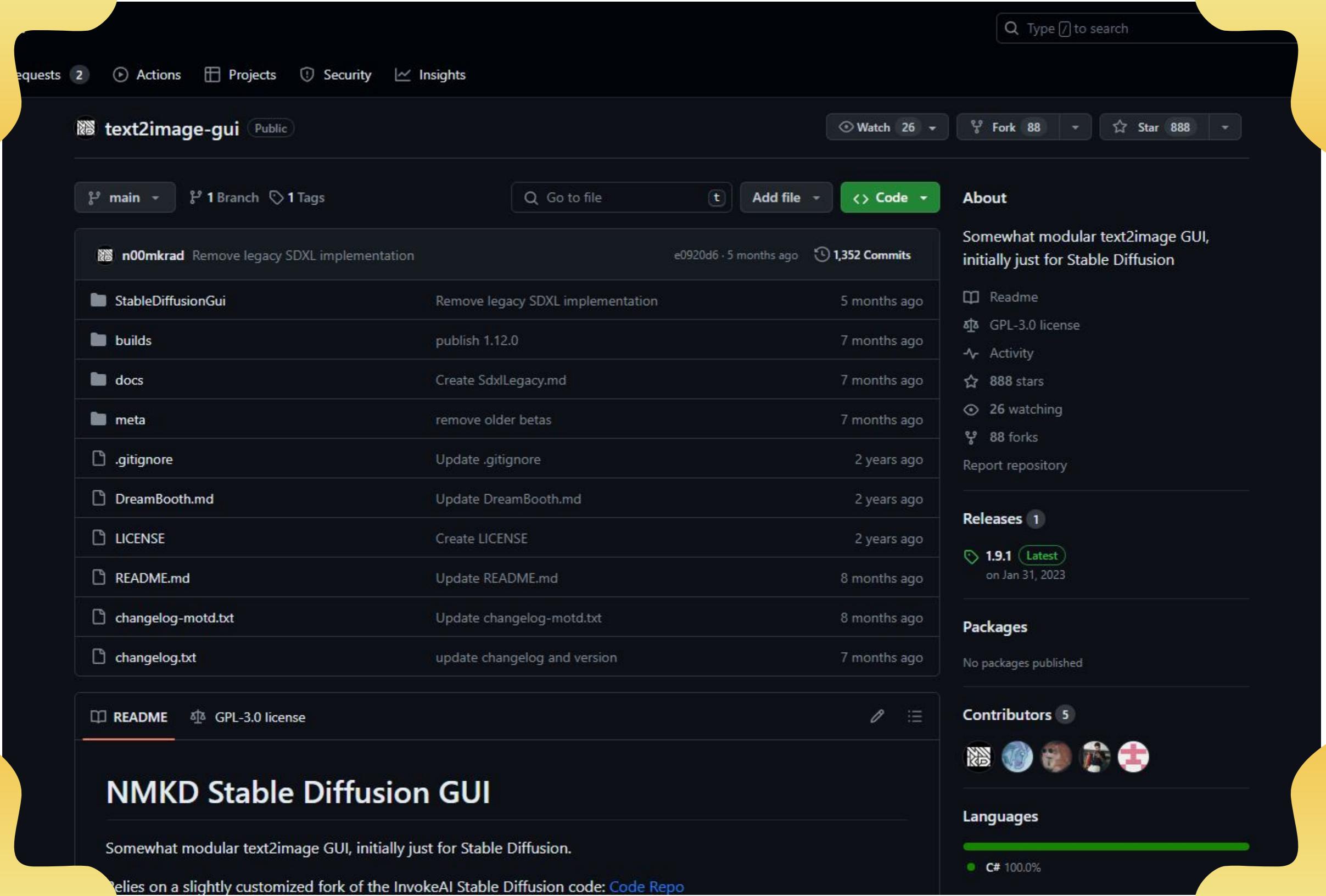
Textual Inversion Embeddings (текстовая инверсия): это небольшие файлы, которые содержат дополнительные концепции, которые можете добавить к вашей базовой модели. Этот концепт может быть: поза, художественный стиль, текстура и т. д. Отрицательные инверсии обучены на нежелательном контенте: можете использовать их в negative prompt, чтобы улучшить изображения.

Lora Files: показывает какие модели LoRA лежат в выбранных папках (так же как и папка для моделей SD, можно выбрать папки для LoRA). Здесь происходит управление ими, можно выбрать несколько LoRA, изменять веса(насколько сильно LoRA будет влиять на генерацию).

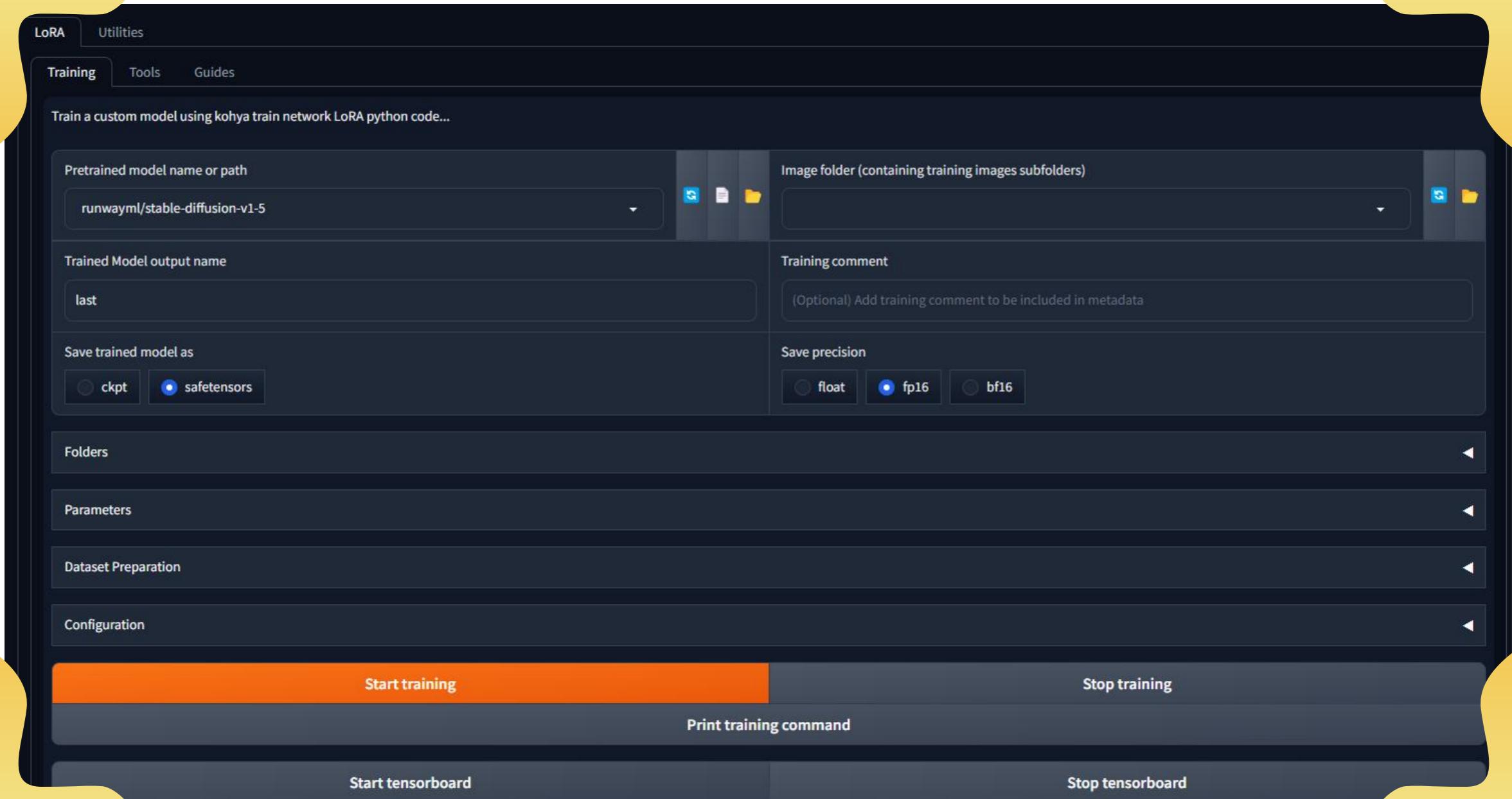
Есть настройка **Sampler**, то же самое что и **Scheduler** в Scenario **c. 25.**

В настройках программы есть еще много различных опций, можно посмотреть и настроить по себе.

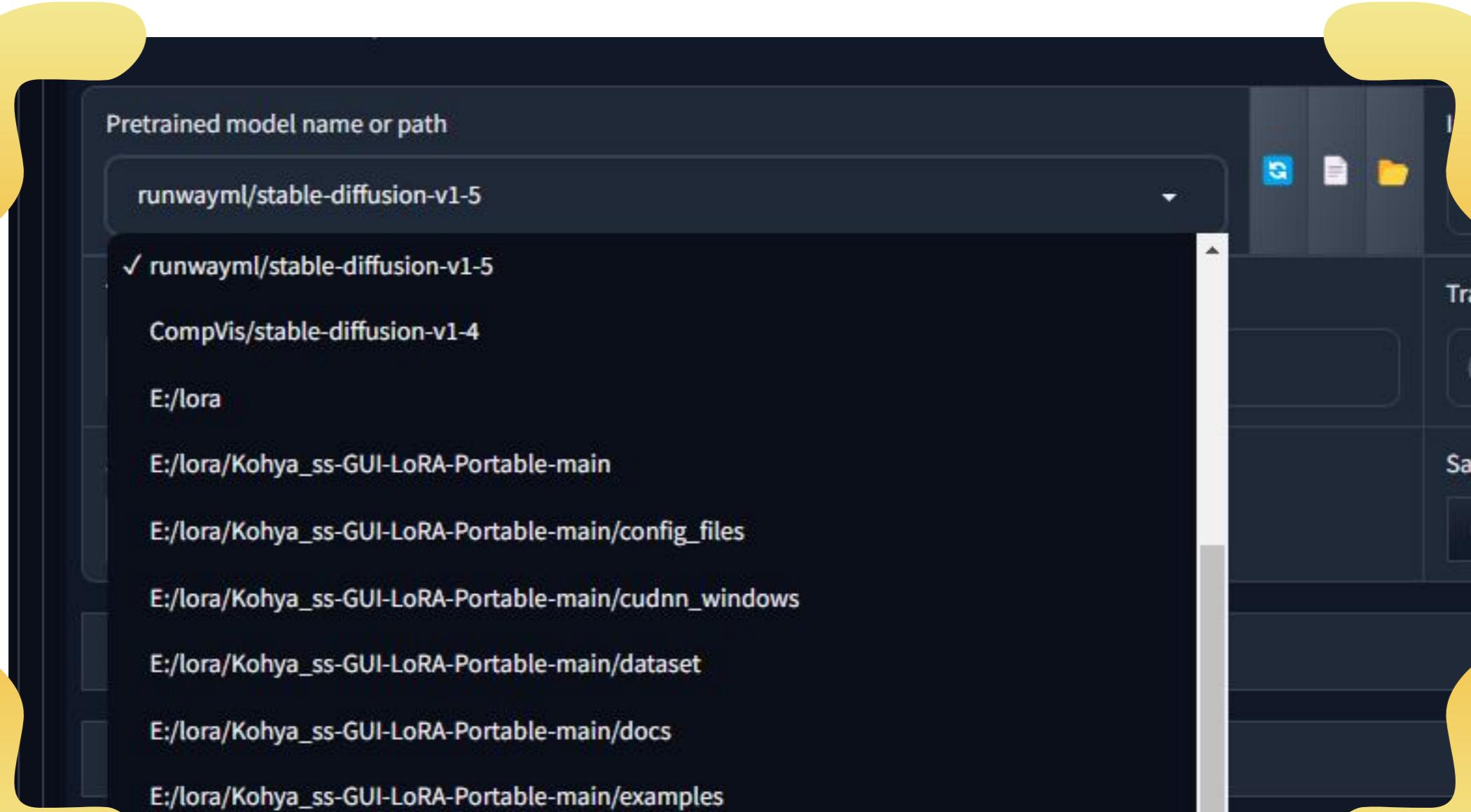
Также с помощью NMKD можно обучать LoRA, инструкция находится на официальном гит репозитории (<https://github.com/n00mkrad/text2image-gui/tree/main>).



Перейдем к **Kohya_ss**. Для тренировки LoRA нужно запускать файл `webui-lora-only`.

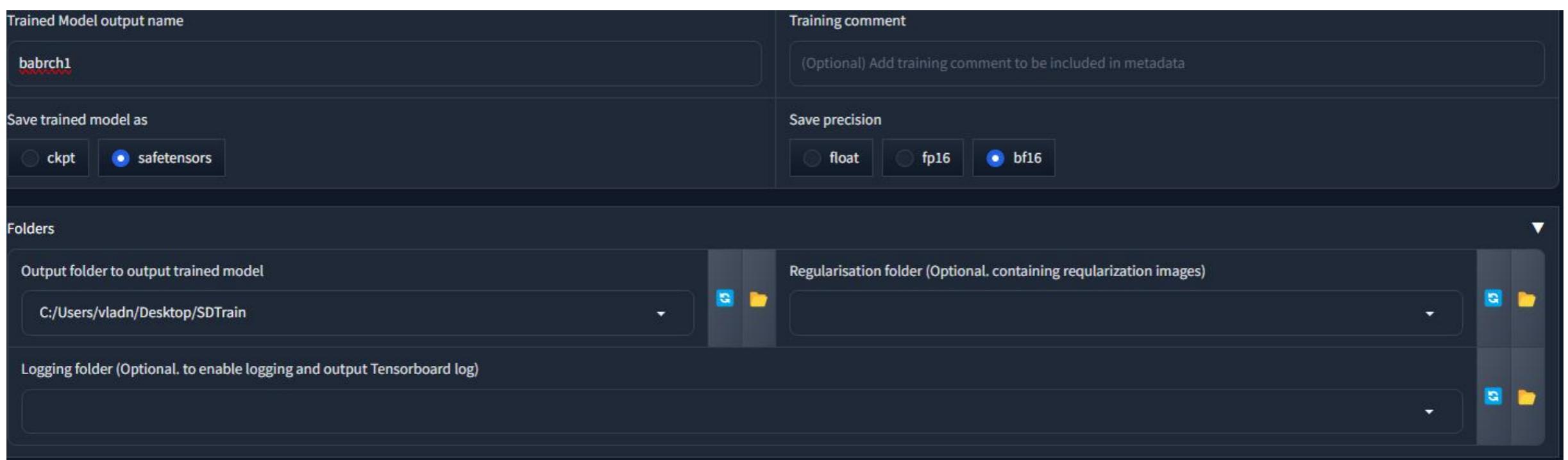


При запуске открывается веб-интерфейс приложения, в котором куча настроек. Рассмотрим только **основные**. Можете посмотреть в интернете, что делают все. Для начала стоит выбрать на какой версии SD будет обучение LoRA. Выбираем из списка либо указываем путь к модели на ПК.



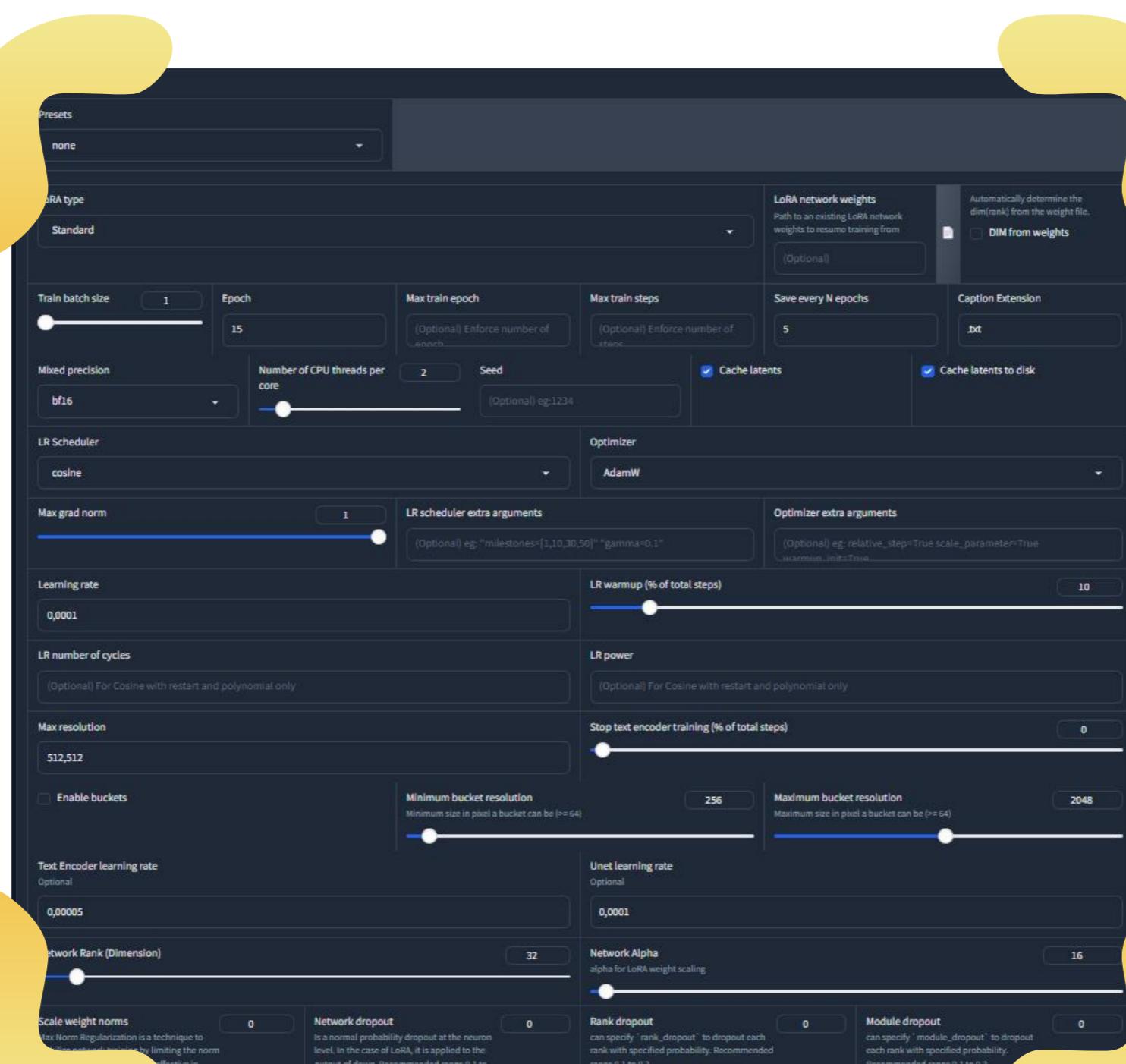
После этого вписываем название LoRA. Опцию **Save trained model as** не меняем, **precision** ставим **fp16**. **fp** - это **floating point (numbers)**, то есть число с плавающей точкой, **16** - число бит, которые выделяем для каждого числа.

Для **bf16** места нужно как для **fp16**, но точность больше, не все видеокарты поддерживают. Установите опытным путем поддерживает ли ваш компьютер такой режим работы.



Выбираем папку в которую будет сохранена модель (**Output folder**).

Далее идет вкладка **Parameters**, в которой очень много настроек, далее будет короткое описание самых важных и базовых.



Train batch size - оставляем 1. Это может ускорить процесс обучения, но влечет за собой большее использование видеопамяти, так как сеть обрабатывает несколько изображений одновременно, усредняя их.

Epoch - сколько эпох. Это кол-во проходов по датасету. Есть два варианта. Оставить тут 1, а папку с датасетом назвать 100_keyword, тогда сеть пройдет по изображениям 100 раз. А можно назвать папку 10_keyword. Поставить 10 эпох. Значит сеть пройдет 10 раз по 10 в итоге снова 100. Эпохи можно сохранять в разделе **Save every N epochs**, поставив например 2. У вас в итоге получится 5 файлов LoRA, каждый из которых натренирован на разном количестве проходов. Это удобный способ избежать недотренированности\перетренированности, просто проверяя все файлы и оставляя лучший.

Number of CPU threads per core - зависит от модели видеокарты, нужно проверить на сайте производителя.

Mixed precision - bf16 по возможности, если нет - fp16

Learning rate scheduler - кривая обучения. В конце руководства ссылка(номер 6) на статью о том, как правильно выбрать. Чаще всего ставят constant для скорости.

Seed - хорошая идея его фиксировать для обучения. Например, сделали LoRA, а она получилась не очень. Как можно быть уверенным что просто не повезло с сидом? Такое бывает. Сид не лег на тренировку. Поэтому можно оставить тот же сид, но поменять параметры. Или наоборот, поменять сид не меняя параметры.



Optimizer - Оставляем по умолчанию. Судя по тестам людей, стандартный показывал себя хорошо. Если не запустится обучение(бывает такая проблема), то ставим любой другой.

Learning rate - чем больше число, тем быстрее обучение, но тем больше деталей упускается. Оставляем по умолчанию.

LR warmup (% of steps) - разминочные проходы на низкой скорости. Оставляем 10 по умолчанию. Для constant ставят 0.

Max Resolution - Если обучающие изображения превышают указанное здесь разрешение, они будут уменьшены до указанного разрешения. Оставляем по умолчанию - 512X512. Многие модели используют изображения такого размера, поэтому лучше всего использовать это. Для обучения SDXL следует использовать 1024X1024.

Enable buckets - даст возможность использовать изображения с неквадратным соотношением сторон, обрезав их. Но лучше готовить датасет нормально. Убираем чекбокс.

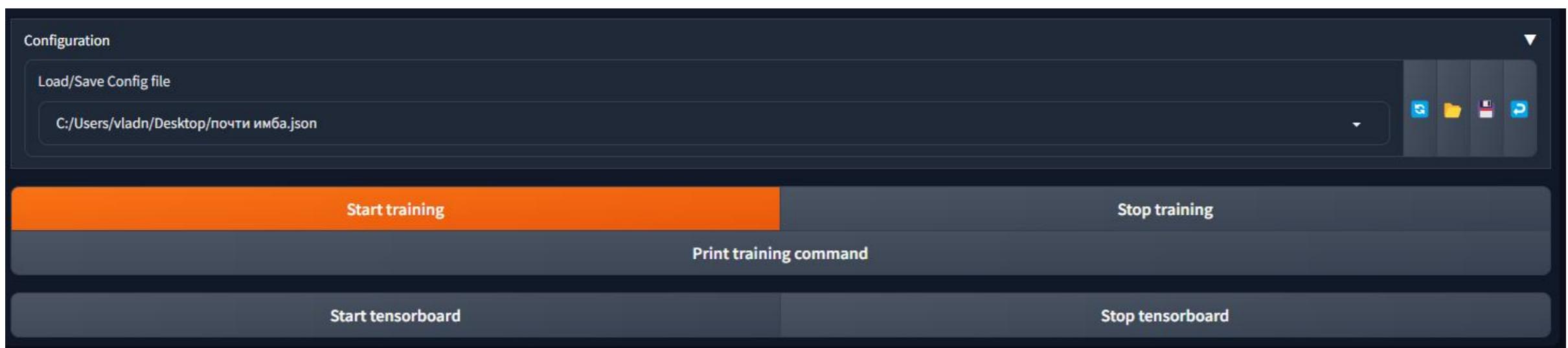
Unet learning rate - что-то вроде памяти, имеет информацию о том как элементы взаимодействуют друг с другом. Первое куда стоит смотреть если обучение не получилось. Сначала рекомендовано пройтись со стандартными параметрами и что-то крутить только когда не получилось. Если итоговая модель выдает визуальный шум вместо нормального изображения то значение слишком высокое. Попробуйте слегка уменьшить и повторить.



Network Rank (Dimension) - повлияет на итоговый размер файла LoRA, примерно 1 за 1 мегабайт. Чем больше число, тем больше обучающей информации может быть сохранено, но увеличивается возможность изучения ненужной информации, отличной от цели обучения. Можно поставить, но если итоговой модели не хватает выразительности стоит попробовать увеличить. **Network Alpha** - это значение должно быть ниже либо равным Network Rank (Dimension) и нужно для предотвращения ошибок точности. Чем ниже значение тем сильнее замедляет обучение. Ставим равным Network Rank.

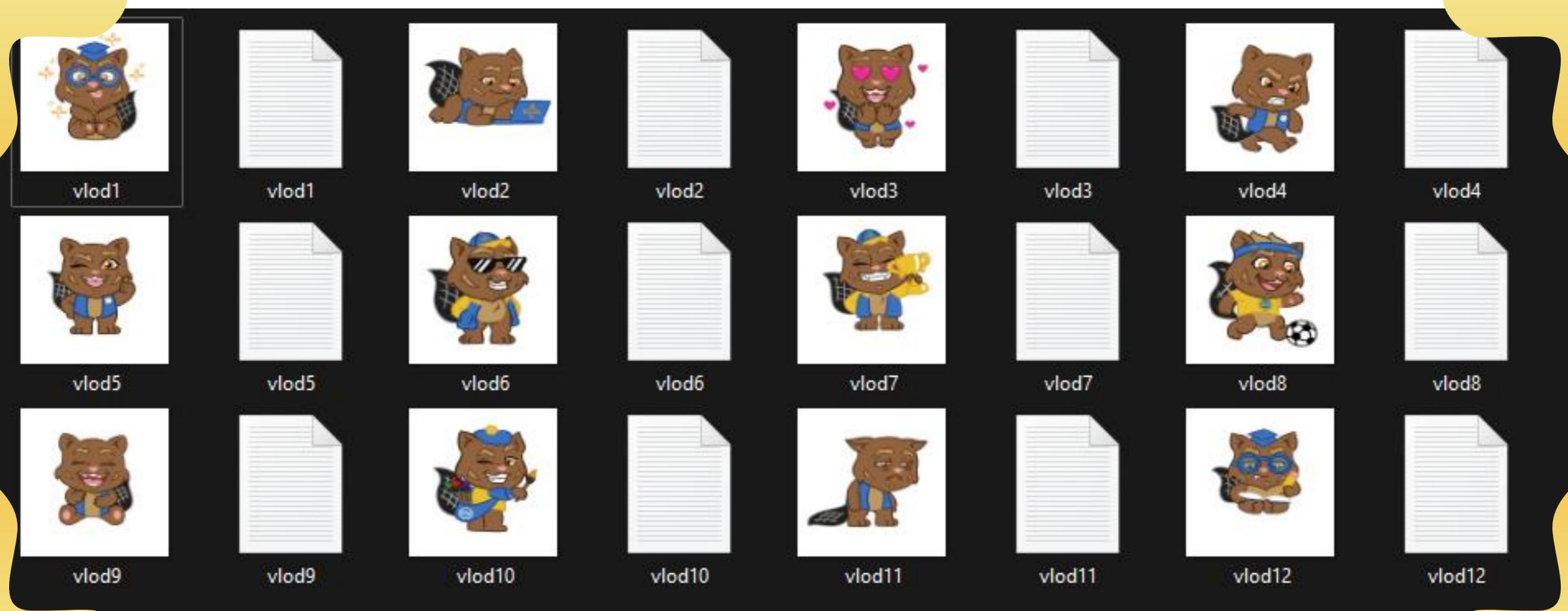
Есть еще куча настроек во вкладке Advanced, но их нужно менять после более глубокого изучения. Если хотите подробно узнать о них посмотрите в интернете или воспользуйтесь ссылками(номер 8)в конце руководства.

Configuration - в данной вкладке можно выбрать готовый файл с конфигурацией обучения, а так же сохранить нынешний с настройками, которые только что выставили.



ПОДГОТОВКА ДАТАСЕТА

Советы всё те же, что на [стр. 14](#), но теперь нужно описывать изображения вручную. Следует подготовить описания для каждой изображения и поместить в папку с изображениями. Легче воспользоваться Scenario и потом уже дополнить описание своими словами. Папка должна выглядеть примерно так: изображения с одинаковыми названиями, но разными цифрами по порядку и подписи к ним в формате txt с аналогичными изображениям названиями.

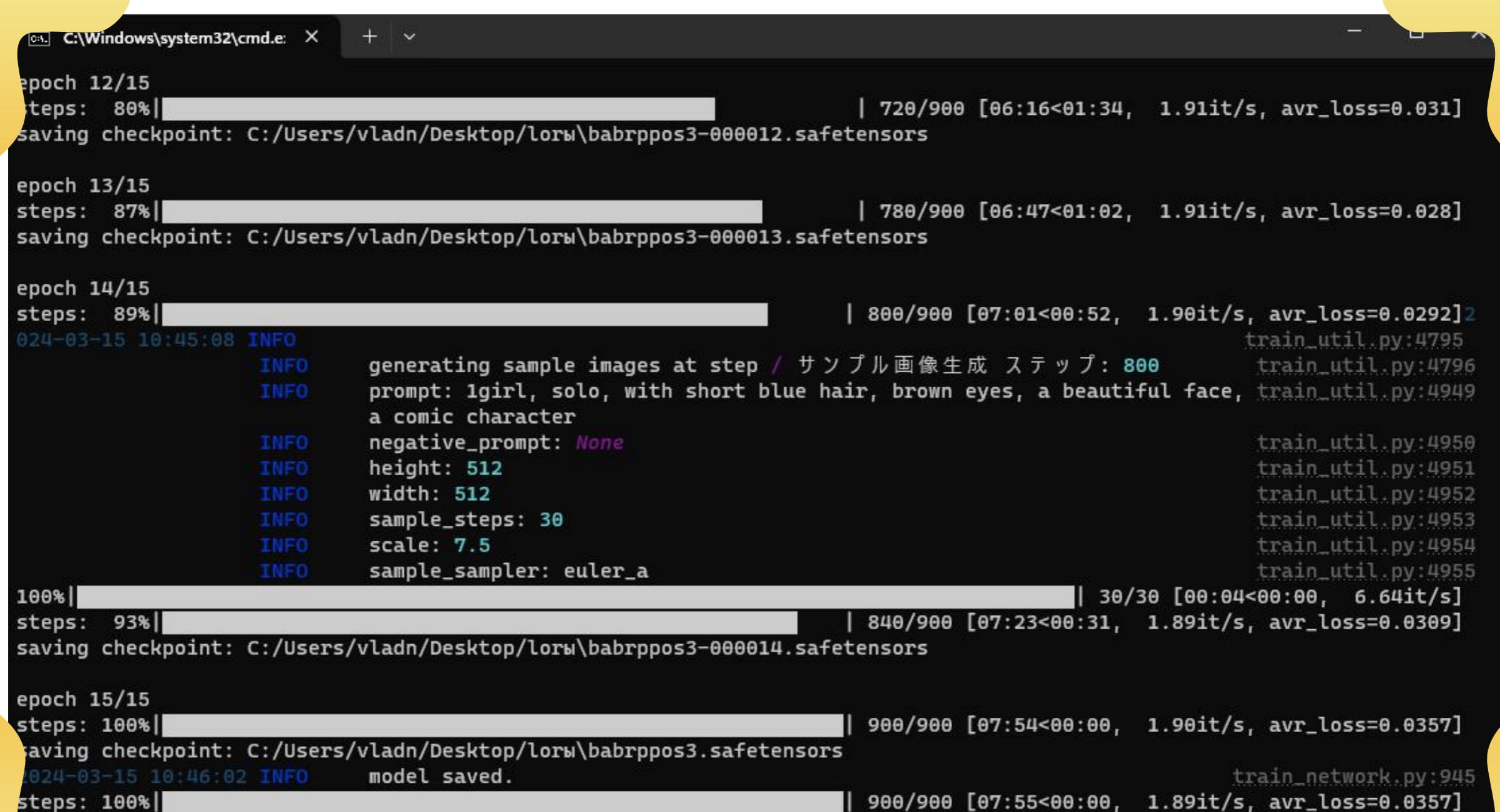


Папка должна иметь название в формате x_babr, где x - количество шагов для обучения. Для примера это 5. В настройках Kohya SS в пункте image folder выбираем папку, В КОТОРОЙ НАХОДИТСЯ папка с изображениями, иначе обучение не пойдет.



РЕЗУЛЬТАТ ТРЕНИРОВКИ

После того как выбрали нужные настройки, нажимаем на Start training и ждем. Скорость обучения будет зависеть от характеристик видеокарты. Чаще всего из-за нехватки видеопамяти обучение может не начаться, поэтому рекомендовано выключить все программы и не трогать компьютер во время обучения.



```
epoch 12/15
steps: 80%|██████████| 720/900 [06:16<01:34, 1.91it/s, avr_loss=0.031]
saving checkpoint: C:/Users/vladn/Desktop/лоры/babrppos3-000012.safetensors

epoch 13/15
steps: 87%|██████████| 780/900 [06:47<01:02, 1.91it/s, avr_loss=0.028]
saving checkpoint: C:/Users/vladn/Desktop/лоры/babrppos3-000013.safetensors

epoch 14/15
steps: 89%|██████████| 800/900 [07:01<00:52, 1.90it/s, avr_loss=0.0292] 2
2024-03-15 10:45:08 INFO      generating sample images at step / サンプル画像生成 ステップ: 800 train_util.py:4795
INFO      prompt: 1girl, solo, with short blue hair, brown eyes, a beautiful face, a comic character train_util.py:4796
INFO      negative_prompt: None train_util.py:4950
INFO      height: 512 train_util.py:4951
INFO      width: 512 train_util.py:4952
INFO      sample_steps: 30 train_util.py:4953
INFO      scale: 7.5 train_util.py:4954
INFO      sample_sampler: euler_a train_util.py:4955
100%|██████████| 30/30 [00:04<00:00, 6.64it/s]
steps: 93%|██████████| 840/900 [07:23<00:31, 1.89it/s, avr_loss=0.0309]
saving checkpoint: C:/Users/vladn/Desktop/лоры/babrppos3-000014.safetensors

epoch 15/15
steps: 100%|██████████| 900/900 [07:54<00:00, 1.90it/s, avr_loss=0.0357]
saving checkpoint: C:/Users/vladn/Desktop/лоры/babrppos3.safetensors
2024-03-15 10:46:02 INFO      model saved. train_network.py:945
steps: 100%|██████████| 900/900 [07:55<00:00, 1.89it/s, avr_loss=0.0357]
```

Когда тренировка окончится, в выбранной папке пункта Output Folder появится файл с моделью. Нам нужно перенести его в папку NMKD\Models\LoRAs.

Открываем NMKD, выбираем LoRA и пробуем генерировать изображения, не забыв использовать выбранное слово для активации модели, в примере это babrich. Для датасета с бабричем результаты приведены ниже. Можно попробовать покрутить различные настройки, чтобы улучшить результат, тут открытое поле для экспериментов.



В любом случае, результаты генераций уже можно использовать как референсы для своей айдентики.

ПОЛЕЗНЫЕ ССЫЛКИ

1. <https://www.youtube.com/@ХриcT> - канал создателя портативной версии Kohya SS, делает видео про подробные настройки тренировки и многое полезное по SD;
2. https://github.com/serpotapov/Kohya_ss-GUI-LoRA-Portable - Kohya SS Portable;
3. <https://nmkd.itch.io/t2i-gui> - NMKD GUI;
4. <https://habr.com/ru/articles/693298/> - как работает SD, объяснение в картинках;
5. civitai.com - сайт с различными моделями SD, LoRA;
6. <https://neptune.ai/blog/how-to-choose-a-learning-rate-scheduler> - как выбрать Learning rate scheduler.
7. <https://stable-diffusion-art.com/samplers> - как выбрать Sampler
8. https://github.com/bmaltais/kohya_ss/wiki/LoRA-training-parameters - о настройках Kohya SS для тренировки LoRA