

負責任的 AI 開發指南



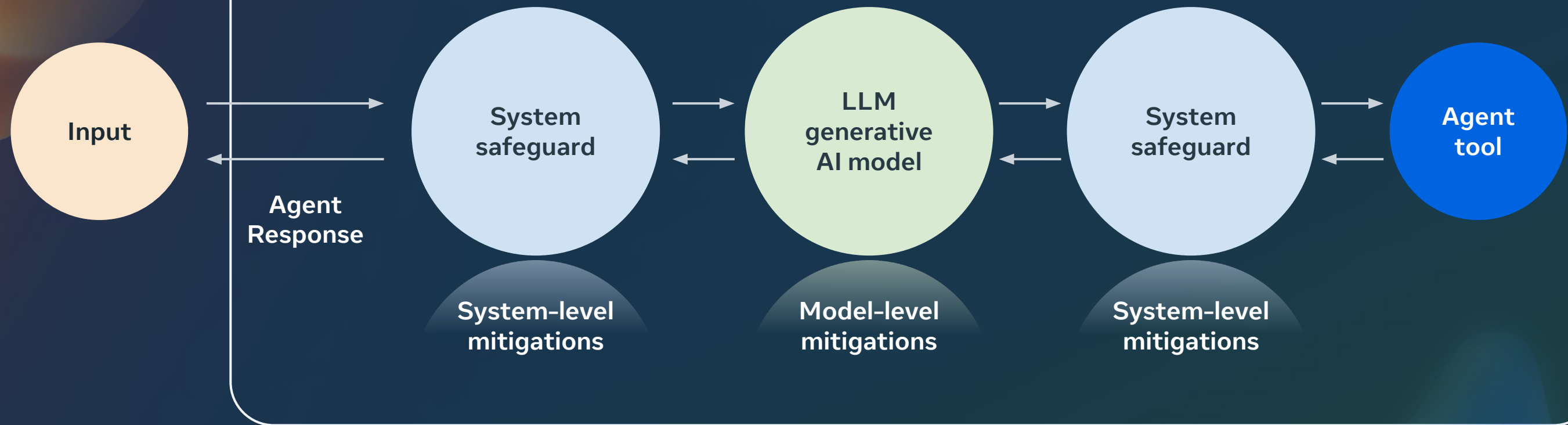
為什麼要遵守這個開發指南？

想像我們正在開發一款新型 AI，如果我們不先確定它的用途，就像是在沒有藍圖的情況下蓋房子，結果可能會完全不符合需求，甚至帶來預期外的風險。

Meta 從過去開發 Llama 的經驗中，提供相關的負責任開發守則給其他開發者參考。

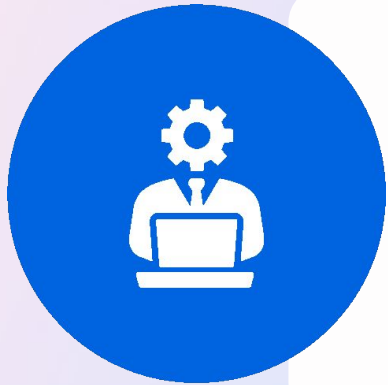


The Llama System



從確定使用案例 (Determine Use Case) 開始

如果我們不清楚 AI 是「要做什麼」和「不該做什麼」，那麼後續的訓練、測試、風險管理就會沒有方向。



這個 AI 是用來做什麼的？

- 醫療建議 AI, 回答高度專業的問題
- 需要確保回答準確且專業



誰是 AI 的使用者？

- 針對企業內部使用的 AI
 - 需要確保它符合企業的數據安全政策，並且能夠提供高精準度的回答。
- 針對一般大眾的 AI
 - 回應就需要更嚴格的監管，避免提供有害或誤導的資訊。

設計內容政策與安全規範

當我們確定 AI 的用途與使用者後，下一個步驟是建立清楚的 內容政策與安全規範，確保 AI 在設計上符合倫理標準，並避免潛在風險。



設計內容政策

內容政策的核心目標是決定 AI「可以」和「不可以」生成的內容。
例如: AI 可以回答如何改善睡眠，但 AI 不可以回答如何駭入別人的電腦。

詳細可以參考ML Commons 制定的分類表。



理解安全與有效回答間的權衡

安全規範機制用來降低 AI 產生不當內容的風險，但也可能因此使 AI 的可用性降低。

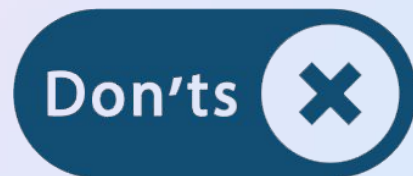
Model-Level Alignment 模型等級的對齊設計

AI 在訓練階段就學到的 內容，以及它在生成回應時的 內建知識與行為。這些限制通常透過 微調(fine-tuning)、數據過濾、RLHF(人類回饋強化學習)或 RLAIIF (AI 回饋強化學習)來實施。



允許 AI 生成哪些內容？

- AI 可以回答「如何改善睡眠習慣？」
- AI 可以提供「2024 年的最新科技趨勢」→ 這些都是 AI 透過訓練學習到的知識點，確保 AI 在回答這類問題時不會產生錯誤資訊。



AI 應該避免哪些內容？

- 不允許 AI 生成非法內容，例如「如何駭入他人電腦？」
- 不允許 AI 提供錯誤的醫療建議，例如「如何自行治療嚴重疾病？」→ 這些需要在訓練階段進行內容過濾，
- 確保 AI 不會學到這些違規資訊。→ 使用 RLHF 或 RLAIIF 來標註 AI 的回應，確保它不會誤提供這類危險內容。

模型負責任的微調流程

(The Responsible Fine-Tuning Flow)

✓ 步驟	🎯 目標	🔑 關鍵措施
準備數據 Prepare Data	確保訓練數據高品質、 具有代表性且符合應用場景	過濾偏見數據、移除個資、 確保數據多樣性
訓練模型 Train the Model	讓 AI 學習特定領域的知識 並保持安全性	使用監督式學習、RLHF/RLAIF、 設定適當訓練參數
評估與改進 Evaluate & Improve	測試 AI 表現, 找出錯誤 並進行修正	自動化測試、紅隊測試、 使用者回饋機制

System-Level Alignment 系統等級的對齊設計

有些內容政策不是透過訓練決定的，而是在 AI 運行時透過 額外的安全機制 來防止不當輸出。
例如，AI 可能仍然「知道」某些資訊，但系統會阻止它提供這些資訊。



輸入與輸出過濾機制

Input & Output Filtering

若使用者輸入：「如何駭入他人電腦？」系統應該能夠即時攔截問題，而不是等 AI 自己決定該不該回答。

即使 AI 產生了一個可能違規的回答，系統應該有能力在輸出階段阻擋，例如使用 Llama Guard 來審查 AI 產出的內容。



內容分級與使用者識別

User-Based Content Restrictions

如果 AI 允許 不同等級的使用者存取不同內容（例如醫療專業人員 vs. 一般使用者），這需要 透過系統層級來管理存取權限，而不是在模型層級解決。



監控與回報機制

Monitoring & Reporting

若 AI 產生了違規內容，系統應該允許使用者回報，並讓開發者持續改進 AI。這部分通常不是模型的問題，而是使用者行為與監督機制的問題，因此屬於 System-Level。

Model-Level vs. System-Level 的總結對比

層級	主要關注點	技術手段	風險管理方式
 模型層級 Model-Level	訓練 AI, 讓它知道該說什麼、 不該說什麼	微調(Fine-tuning)、RLHF、 RLAIF、Red Teaming	在訓練階段調整 AI 的行為, 讓它學習正確回應
 系統層級 System-Level	控制 AI 的使用方式, 防止被濫用	輸入過濾、輸出監測、 用戶回報機制、透明度機制	監測 AI 的運作情況, 確保它不會產生不當內容

四大關鍵領域的負責任 AI 開發準則

1 Image Reasoning 影像推理

主要考量

AI 需理解影像與文字的結合，並確保輸出不產生錯誤或違法內容。

具體解決方案：

- 使用 Llama Guard-Vision 來過濾影像輸入與輸出，防止不當內容。
- 限制 AI 不可識別影像中的個人資訊，以避免隱私洩露。
- 針對影像中的提示語 (Prompt Injection) 進行安全防護，防止惡意攻擊。



四大關鍵領域的負責任 AI 開發準則

2 Multilinguality 多語言支持

主要考量

確保 AI 在不同語言環境下的表現準確，並避免因語言或文化差異產生誤解或偏見。

具體解決方案：

- 測試 AI 在不同語言的表現，確保語言支持的完整性，避免低資源語言的錯誤。
- 針對不同語言設計敏感詞過濾機制，防止 AI 產生文化冒犯或不適當內容。
- 調整內容審查標準，確保 AI 在各語言環境下符合當地法律與文化規範。
- 優化系統級安全機制，如 Llama Guard，擴展對多語言的支持，並透過語言過濾器限制 AI 存取未經安全調校的語言，防止產生潛在風險內容。



四大關鍵領域的負責任 AI 開發準則

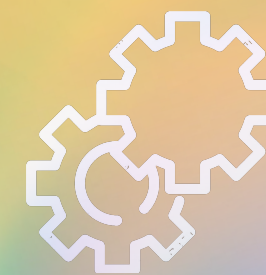
3 Tool Calls 工具調用

主要考量

確保 AI 透過 API 連接外部工具時，能夠安全使用，並避免調用惡意或不信任的服務，防止 LLM 被濫用或遭到攻擊。

具體解決方案：

- 部署適當的系統安全機制，例如使用 **Llama Guard** 來審查來自第三方工具的輸出，確保不含違規內容。
- 強化 API 輸入與輸出監控，防止外部工具傳送惡意請求給 LLM，或 LLM 產生有害查詢給工具。
- 使用 **Prompt Guard** 偵測 **LLM Jailbreak 攻擊**，防止透過繞過限制的方法讓 LLM 執行未授權的操作。
- 在程式碼執行環境中增加額外的安全機制，例如使用 **Llama Guard** 來限制 LLM 生成惡意程式碼，避免程式碼解釋器 (code-interpreter) 執行有害代碼。
- 檢查 LLM 是否支持適當的工具調用，參考 **Llama 3.2** 的模型文件，確保 AI 只會執行預先定義的安全 API 調用。
- 對於非文字類型的輸出，部署額外的防護措施，例如對於影像或音訊輸出，增加審查與過濾，以防止違規內容。



四大關鍵領域的負責任 AI 開發準則

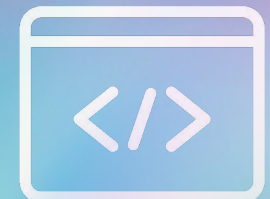
4 Coding 程式碼生成

主要考量

確保 AI 產生的程式碼安全、符合最佳實踐，並避免生成惡意或含有安全漏洞的代碼。

具體解決方案：

- 部署適當的系統安全機制，例如使用 **Llama Guard** 來審查來自第三方工具的輸出，確保不含違規內容。
- 強化 API 輸入與輸出監控，防止外部工具傳送惡意請求給 LLM，或 LLM 產生有害查詢給工具。
- 使用 **Prompt Guard** 偵測 **LLM Jailbreak 攻擊**，防止透過繞過限制的方法讓 LLM 執行未授權的操作。
- 在程式碼執行環境中增加額外的安全機制，例如使用 **Llama Guard** 來限制 LLM 生成惡意程式碼，避免程式碼解釋器 (code-interpreter) 執行有害代碼。
- 檢查 LLM 是否支持適當的工具調用，參考 **Llama 3.2** 的模型文件，確保 AI 只會執行預先定義的安全 API 調用。
- 對於非文字類型的輸出，部署額外的防護措施，例如對於影像或音訊輸出，增加審查與過濾，以防止違規內容。



開發整合 AI 工具，提供服務的負責任重點



整體最佳化 Holistic Optimization

AI 開發的每個階段相互影響，過度優化單一環節可能會導致其他部分的性能下降。例如，過度過濾訓練數據雖可提升安全性，但可能會降低模型處理不安全內容的能力。因此，應在整個開發生命週期中佈署不同層級的安全機制，確保 AI 既安全又高效。



確保開發各階段的目標一致 Alignment of Objectives

從數據收集到使用者回饋，每個階段都應該與最終應用目標保持一致。確保團隊在開發過程中遵循統一的安全與性能標準，避免在不同階段出現相互矛盾的優化方向。



標準化回饋學習機制 Standardizing Learning from Feedback & Errors

建立明確的流程來收集並分析使用者回饋與錯誤，確保模型能夠持續改進。這包括回饋分析、問題優先排序，以及將學習成果系統性地應用於下一輪模型訓練，使 AI 在不斷演進的環境中保持可靠性。

生成式 AI 技術快速發展，開發者需持續學習其影響與風險，
並強調**透明度、責任感與使用者賦能 (User Empowerment)**。

透過不斷調整與改進，讓我們一起在使用與開發 AI 的路途中，
既創新，又符合社會責任。