

【解説】 生成AIの世界モデル

生成AIを実際の社会で活用する場合、たとえば自動運転や工場の自動化などに応用しようとする
と、今の大規模言語モデル（LLM）のままではうまく機能しません。

これは、ネット上の知識だけでなく、現実世界の知識までモデルに取り込む必要があるからです。
この辺りの話は、「生成AI活用フォーラム 2025 夏」で東京大学 松尾・岩澤研究室の鈴木雅大氏が
講演されていた内容にも通じます。

先に書いた講演会の報告書でも触れましたが、今回は生成AIにおける「世界モデル」について
レポートしたいと思います。

「世界モデル」という言葉は少し分かりにくいですが、簡単に言えば、現実世界で自動運転などに
使えるような、より拡張された知識の枠組みを作ることです。

ただ、この考え方や技術は、専門外の間がいきなり説明するのはかなり難しい。そこで、前回の
生成AIの解説のときと同じように、今回も生成AIの助けを借りることにしました。

ただ今回は、前に使ったChatGPTではなく、Copilotに頼ってみることにしました。前回同様、質問
を青で囲んで、その下にCopilotの回答を書く形で示し、**世界モデルがどういうものか、何が難しい
のかを理解することをを目標**とします。また、**レポートの範囲は自動運転に注目し、ポイントが
ボケないように動画生成などの分野は省きます**。また、回答も**専門的な内容は省略し、筋を追えば、
なんとなく理解したつもりになれるレベル**を目指します。

ただし、このHPに掲載されている【生成AIとは何か（1） - （4）】を事前に読んでおいてくださ
い。LLMが次の言葉をどの様に選ぶのか、その仕組みについては理解しておきましょう。

【質問】

端的に、生成AIの世界モデルとはどんなものですか？

【Copilotの回答】

🗨️ 世界モデル（world model）とは、生成AIが「世界をどう理解しているか」を表す内部構造です。
つまり、現実世界の構造や関係性を抽象的・数理的に表現したモデルであり、AIが予測・創造・推
論をするための土台になっています。

🔗 端的に言えば…

生成AIの世界モデルとは、「観測されたデータをもとに、環境の構造や因果関係を内部で再構築す
る仕組み」です。

【つぶやき】

良く分からない説明ですが、【環境の構造や因果関係を内部で再構築するをするための土台】と言
うことは、ChatGPTと同じように、計算で因果関係を見つけられるようなデータの塊（クラスター）
を作ると言う事の様です。

【質問】

生成AIの世界モデルはChatGPTのような大規模言語モデルと何が違い、何が同じなのでしょう
か？

【Copilotの回答】

以下に、世界モデルと大規模言語モデル（LLM）の「共通点」と「相違点」を端的にまとめまし
た。

☑️ 共通点

予測ベースの学習：どちらも次の状態や単語を予測する訓練を受けています。

内部的な表現空間：抽象的な構造や意味を埋め込んだベクトル空間を活用しています。

自己教師あり学習：外部ラベルなしで、大量のデータからパターンを抽出しています。

🔍 相違点		
項目	世界モデル	大規模言語モデル（ChatGPTなど）
目的	環境や文脈の「状態遷移」を理解・シミュレーション	自然言語を生成・理解する
対象構造	状態空間・因果関係・物理モデルなど	言語構造・文脈・統計的パターン
出力形式	状態予測・行動方針・環境シミュレーション	テキストの生成・要約・翻訳など
利用領域	ロボティクス、強化学習、思考の推論	対話、執筆支援、情報検索など
因果性の扱い	明示的にモデリングされることが多い	多くは統計的関連性に依存

要するに、世界モデルは「何が起こるか」を理解しようとする頭脳であり、LLMは「何を言うか」を得意とする語り部です。

【つぶやき】
端的に言えば、世界モデルは次に何が起きかを予測するモデル構造であり、大規模言語モデルは次にどんな言葉が来るかを予測するモデル構造だということです。
モデルと言うと雛形と言うイメージを持ちますが、ここではデータ群で作った、現実世界もどき、敢えて言えば、人体のスケルトン模型をイメージしたらよいと思います。

【質問】
生成AIの世界モデルはChatGPTのような大規模言語モデルと技術的には何が違い、何が同じなのでしょう？

【Copilotの回答】
技術的な観点から「生成AIの世界モデル」と「ChatGPTのような大規模言語モデル（LLM）」の共通点と違いを端的に整理します。
☒ **技術的に同じ点（共通基盤）**
●**Transformerアーキテクチャ**
→ 両者ともに「Attention Is All You Need」で提案されたTransformerを基盤にしています。
●**自己教師あり学習（Self-supervised Learning）**
→ ラベルなしの大量データからパターンを抽出する学習方式を採用。
●**ベクトル空間による意味表現**
→ 単語や状態を高次元ベクトルで表現し、意味や関係性を捉えます。
●**確率的予測モデル**
→ 次の単語（LLM）や次の状態（世界モデル）を確率的に予測する構造。

🔍 技術的に異なる点（設計思想と構造）		
項目	世界モデル（生成AI）	大規模言語モデル（ChatGPTなど）
対象領域	状態空間・環境・行動	自然言語・文脈・語彙
出力形式	状態遷移・行動方針・環境シミュレーション	テキスト生成・要約・翻訳
因果性の扱い	明示的にモデリング（例：物理法則やエージェントの行動）	統計的関連性に依存（暗黙的な因果性）
学習データ	センサー情報、環境シミュレーション、強化学習履歴など	大量のテキスト（書籍、Web、会話など）
応用分野	ロボティクス、自律エージェント、ゲームAI	対話、文章生成、検索、教育支援

●つまり、世界モデルは「環境の理解と予測」に特化した構造であり、LLMは「言語の生成と理解」に特化した構造です。**両者はTransformerという共通の技術基盤を持ちながら、対象とする世界の構造が根本的に異なる**のです。

【つぶやき】
なかなか、端的に答えています。技術基盤は共通ですが、対象とするデータ群の構造は違うと言う事らしいです。

【質問】
生成AIのLLMでは文脈ベクトルは300から500次元になると聞いています。世界モデルになると、文脈ベクトルに相当する世界モデルの表現ベクトルの次元はどのくらいになりますか？

【Copilotの回答】
LLMと世界モデルにおけるベクトル次元の違いを技術的に整理します。
🗣️ LLM（大規模言語モデル）の文脈ベクトル
一般的なLLM（例：GPT-2, GPT-3）では、**文脈ベクトルの次元数は300～4096程度**。
OpenAIのtext-embedding-ada-002では1,536次元のベクトルが生成されることが確認されています。
文脈ベクトルは、主に語彙・文脈・意味的関係性を捉えるための抽象空間です。
🌐 世界モデルのベクトル次元（状態表現）
世界モデルでは、状態空間の表現ベクトルが文脈ベクトルに相当します。
例えば、MetaのV-JEPA 2では、**数千～数万次元の抽象ベクトル空間が使われており**、物理的状态や因果関係を表現します。

🔍 比較表			
モデルタイプ	ベクトルの次元数	表現対象	備考
LLM（GPT系）	約300～4096次元	言語文脈・意味	Transformerの隠れ層に依存
Embeddingモデル（OpenAI）	1,536次元	意味的類似性	text-embedding-ada-002など
世界モデル（V-JEPA 2など）	数千～数万次元	状態・物理・因果	映像・動作・環境を統合的に表現

つまり、世界モデルのベクトル空間は、LLMよりも遥かに高次元かつ構造的であり、因果性・物理法則・時間的变化などを捉えるために設計されています。
この違いは、単なる次元数の差ではなく、空間の意味論的構造の違いに直結しています。

【つぶやき】
大規模言語モデルと世界モデルの違いは基本的にデータセットの違いだということです。どの位、違うのかと言うと、大規模言語モデルの次元数がせいぜい数千なのに対し、世界モデルは数万にも及ぶということです。数万次元のベクトルなどは想像もつきませんが、現象を修飾するパラメータがそれだけあるという程度に思えば十分でしょう。

【質問】
LLMとは比べものにならない計算パワーが必要になりそうですが、世界モデルの生成AIを自動車に積めるまでコンパクトに構成できますか？

【Copilotの回答】
結論から言うと、世界モデルを自動車に搭載できるほどコンパクトに構成することは技術的に可能ですが、いくつかの工夫と制約が伴います。
🚗 世界モデルを車載可能にするための技術的工夫

以下のようなアプローチが現実的に検討されています：

- モデルの量子化と圧縮
→ 精度を保ちつつ、計算量とメモリ使用量を削減する技術。INT8量子化などが代表例です。
- モジュール分割とエッジ最適化
→ 世界モデルの一部（例えば短期予測や局所判断）を車載チップで処理し、長期予測や複雑な推論はクラウドや外部サーバで補完。
- 専用ハードウェア（例：NVIDIA Jetson, Tesla FSD Chip）
→ 高性能なGPUやTPUを車載向けに最適化したチップが登場しており、リアルタイム推論が可能になりつつあります。
- 低レイテンシ設計
→ 推論速度を最優先に設計された軽量モデル（例：Distilled Transformer）を活用。

LLMとの比較：計算負荷の違い			
モデルタイプ	推論負荷	主な処理対象	車載化の難易度
LLM（GPT系）	中～高	言語・文脈	可能（既に搭載例あり）
世界モデル（GAIA-1など）	高～非常に高	状態遷移・物理・因果	部分的に可能（工夫が必要）

※ GAIA-1のような世界モデルは、動画・テキスト・行動を統合的に処理するため、**コンテキスト長が16K以上、トークン数が1万超**というケースもあり、現状ではフルモデルの車載は困難。¹

つまり、世界モデルのフルスペックをそのまま車に積むのは現時点では非現実的ですが、**必要な機能を抽出・圧縮し、リアルタイム判断に特化した構成にすれば十分に実用可能です。**

【つぶやき】
つまり、まともに取り組むと自動車に積むには非現実的だが、機能を分けて、リアルタイムに処理する部分は自動車に載せ、そうでない部分はネットでつなぐような構成にすればできる、という事のようにです。

【質問】
自動運転における生成AIの世界モデルはどの程度の完成度になれば使える技術と判断できるのでしょうか？

【Copilotの回答】
生成AIによる世界モデルが「使える技術」と判断されるには、単なる精度や性能だけでなく、安全性・汎用性・リアルタイム性・**解釈可能性**など、複数の観点で一定の完成度を満たす必要があります。以下に、技術的・実用的な判断基準を整理してみます。

✔ 技術的完成度の判断ポイント		
観点	要件	実用化の目安
予測精度	歩行者・車両・環境の未来状態を高精度に予測	事故率が人間以下になること
リアルタイム性	車載チップで数百ms以内に推論可能	推論レイテンシが300ms未満
汎用性	未知の状況（エッジケース）にも対応	動物・工事・異常気象なども予測可能
解釈可能性	判断根拠を説明可能（XAI）	規制当局が安全性を検証できる
学習データの多様性	地域・文化・交通ルールの違いを網羅	数千時間以上の走行データが必要
安全性の証明	ハルシネーションや誤判断の抑制	RSS（責任感知安全）などの形式的検証が可能

🧠 判断基準の哲学的側面
「因果性」や「情報幾何」の観点から見ると、世界モデルが物理的・社会的因果関係を低次元で抽象

化できるかが鍵になります。つまり、**単なる統計的予測ではなく、意味のある構造的理解ができるかどうか**。

この点で、生成AIが「使える技術」となるには、以下のような進化が必要です：

- 因果グラフの埋め込みによる**行動予測の透明性**
- 情報幾何による状態空間の圧縮と可視化
- 黒意思決定における**反実仮想（counterfactual）シミュレーション**

つまり、世界モデルが「使える技術」になるには、単なる精度ではなく、**構造的理解と安全性の両立**が必要です。

【つぶやき】
この辺りは生成AIの根本にかかわる非常に重要な論点です。単に事故率が今より下がるというだけでは使えないと言ってます。
何故、そう判断したのか？ そうしなかった場合にどうなると予想したのか？ などがある程度明確に説明できないと、実際に検証に耐えられる、使える技術にはならないと言ってる訳です。

【質問】
こうした現状を踏まえ、日本で自動運転が実用化されるにはいつ頃になりそうですか？

【Copilotの回答】
日本ではすでに一部の地域で自動運転レベル4のサービスが開始されており、政府は2025年度までに50カ所、2027年度までに100カ所以上での展開を目指しています。
この目標は、地域限定型の無人自動運転移動サービスに関するもので、都市部や地方の公共交通の補完としての役割が期待されています。

🚦 実用化のフェーズと予測タイムライン		
年度	実用化の段階	主な内容
2025年	先行導入期	レベル4サービスが10～20地域で稼働。高速道路での実証も進行 ¹ ²
2027年	拡大期	100地域以上でのレベル4サービス展開。物流やバス運行での活用が本格化 ¹
2030年以降	本格普及期	高速道路でのレベル4トラック運行や、都市部でのロボタクシー導入が進む ³
2040年以降	レベル5の可能性	完全自動運転（全条件下）への移行が技術的に視野に入る ³

つまり、2025～2027年にかけては「限定的な実用化」、**2030年以降は「本格的な社会実装」**が始まると見られています。世界モデルや因果推論の応用も、こうしたフェーズで重要な役割を果たすでしょう。

【総括】
自動運転に絞って、生成AIの世界モデルの現状についてレポートしました。生成AIは今後どんどん我々の社会に入り込んでくることは間違いありません。しかし、そうなった時に、今回のレポートで問題になった事柄が、自動運転の世界だけではなく、教育や医療はたまた行政、政治の世界でも重要な課題になることを忘れてはならないでしょう。

以上