

Present but Barely Used: Reproducing the Emergent Othello World Model in a Masked Diffusion Language Model

Éric Jacopin

Cosmic AI

eric.jacopin@protonmail.com

Abstract

Autoregressive (AR) sequence models trained only to predict legal Othello moves develop an emergent, linearly-decodable representation of the board in a player-relative (MINE/YOURS) frame (Li et al., 2023; Nanda et al., 2023). We ask whether this emergent linear world model survives a change of *training objective*: from next-token prediction to *masked diffusion* (Sahoo et al., 2024). On an identical GPT-2 backbone and identical data, we train an AR control and a masked diffusion language model (MDLM), reproduce on the control the linear/relative dissociation of Li et al. (2023) and Nanda et al. (2023) (henceforth the Li–Nanda dissociation), and probe the MDLM with the same battery. We find that the linear player-relative world model *transfers* to masked diffusion (the dissociation reproduces) but is *substantially weaker* (~ 0.79 vs. the AR’s 0.96 two-way accuracy). A causal intervention (Li et al., 2023; Nanda et al., 2023) shows the representation is only *minimally used*: a statistically real but tiny and inconsistent effect (mean $+0.02$, 24% win-rate) against the AR’s strong, consistent $+0.21 / 84\%$. Order-artifact controls confirm the representation is genuine rather than an artifact of an easy unmasking order; both gaps persist across the difficulty strata of the game, so they are not a disguised competence gap, and across three training seeds per objective. As a reproducibility contribution, this is a generalizability stress test of the Li–Nanda finding under a change of training objective, anchored throughout by random-weight and shuffled-label control tasks and by component ablations. We report a second implementation, in Rust on an independent interpretability stack, that reproduces our headline numbers. Code, data recipes, and model checkpoints are publicly available (Jacopin, 2026b).

1 Introduction

A striking result in interpretability is that a transformer trained only to predict legal moves of Othello

builds an internal model of the board: Li et al. (2023) recover the board state with nonlinear probes, and Nanda et al. (2023) show, at BlackboxNLP 2023, that the representation is in fact *linear* once read in a *player-relative* (MINE/YOURS) frame rather than an absolute (black/white) one. This “emergent linear world model” has become a touchstone for mechanistic interpretability of sequence models.

Nearly all of this work studies *autoregressive* (AR) models. A different and increasingly prominent objective is *masked diffusion* (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024), in which a model denoises a partially-masked sequence bidirectionally. Masked diffusion language models (MDLMs) match AR perplexity on language while decoding in arbitrary order. Does the emergent linear world model depend on the autoregressive objective, or does it survive the switch to masked diffusion?

We answer this for Othello. Holding the backbone (a GPT-2-style network) and the training data (20M synthetic random-legal games) fixed, we train an AR control and an MDLM whose *only* difference is the objective, and probe both. Our contributions are:

1. We reproduce the emergent linear world model of Li et al. (2023) and its linear/relative *dissociation* (Nanda et al., 2023) (the Li–Nanda dissociation) on our own AR pipeline, and characterise the gap to the original (~ 0.96 vs. >0.99) as a genuine model-fidelity difference rather than a probing artifact (§4).
2. To our knowledge, we give the first probe of an emergent board world model in a *masked diffusion* sequence model (denoising encoder–decoders such as T5 and Bart have been probed, Yuan and Søgaard, 2025, but not a diffusion-trained generator). The linear player-relative world model *transfers* but is *substan-*

tially weaker (§4).

3. A causal intervention shows the MDLM’s board is *present yet barely used*: a presence-vs-use dissociation measured against ground truth (§5).
4. We show both gaps are robust: they persist across the difficulty strata of the game, so they are not a disguised competence gap, and across three training seeds per objective (§6); and we rule out that the representation is an artifact of an easy unmasking *order* (Trainin et al., 2026), on both the probe and the intervention (§7).
5. We reproduce the headline probe and intervention results in a fully independent second implementation (§8), a strong reproducibility control.

Relation to the reproducibility challenge. In the challenge’s terms, this paper is primarily a *generalizability* study: does the Li–Nanda finding survive a change of training objective, model direction (causal $\rightarrow$ bidirectional), and decoding process? It is anchored by strong *baselines* (random-weight and shuffled-label control tasks, Hewitt and Liang, 2019), by targeted *ablations* (the probe’s basis, functional form, and layer; the input masking regime; the read order; the edit strength), by *complementary evaluation* (a causal-use metric beyond probe accuracy), and by *statistical controls* (difficulty stratification and three training seeds per objective). Rather than rerunning the released codebase, we reproduce the original finding with a fully independent pipeline, then reproduce our own results in a second, *tooling-environment* generalization (the Rust stack of §8): a deliberately stronger setting, since it validates the finding and our own pipeline at once. We report a partial negative result (the diffusion world model is present but barely used) as a boundary condition that enriches rather than questions the original claim.

Paper outline. Section 2 reviews the Othello world-model and masked-diffusion literature and the probing pitfalls that motivate our controls. Section 3 describes the rule engine and oracle, the backbone- and data-matched AR and MDLM models, and the probe battery. Section 4 shows the linear world model transfers to masked diffusion but is substantially weaker; Section 5 shows it is barely used; Section 6 shows both results are robust to position difficulty and to training seeds; and Section 7 rules out a decode-order artifact. Section 8 reports the independent reimplementations, before

the conclusion and limitations. Appendix A collects every experimental parameter; Appendix B the full per-layer and per-stratum tables.

2 Background and Related Work

Emergent world models in Othello. Li et al. (2023) train OTHELLO-GPT on move sequences from random-legal games and recover the board with nonlinear probes; interventions edit the board representation and steer predictions. Nanda et al. (2023) show the representation is linear in the MINE/YOURS frame, where each occupied cell is labelled *mine* or *yours* relative to the side to move. Follow-up work probes and revisits these claims (Hazineh et al., 2023); Yuan and Søgaard (2025) extend the grounding result to seven model families, including denoising encoder–decoders (T5, Bart), with up to 99% accuracy and highly similar learned board features. Related state-tracking probes appear in chess (Toshniwal et al., 2022). Linear probing itself follows Alain and Bengio (2016).

Probing pitfalls and controls. Probing is not self-validating: a probe can fit structure even in a randomly-initialised network (Hewitt and Liang, 2019; Méloux et al., 2025), and a linear probe can exploit surface textual leakage rather than a genuine representation (Belinkov, 2022; Boxo et al., 2025). The discipline is to bound every probe with *control tasks* (random-weight and shuffled-label baselines) and, better still, to complement decoding with a *causal* test that the representation is actually used. We do both throughout, and note that the Othello board is a *latent* quantity: it is recoverable only by simulating the rules from the move sequence, never by reading the surface tokens, which itself guards against surface leakage.

Masked diffusion language models. Discrete diffusion (Austin et al., 2021) corrupts tokens toward an absorbing state and learns to denoise. Sahoo et al. (2024) give a simple, effective masked-diffusion objective (MDLM) with a Rao–Blackwellised evidence lower bound that reduces to a reweighted per-masked-token cross-entropy; we use exactly this objective. Unlike AR models, an MDLM can unmask positions in *any* order, which raises a confound for world-model claims: a model might solve a state-tracking task by exploiting an easy decoding order rather than by holding a rich internal state (Trainin et al., 2026). We address this confound directly in §7.

3 Experimental Setup

Engine and oracle. We reimplement the Othello rules from scratch as a dependency-free engine that yields, for every partial game, the ground-truth board, the legal-move set, and the side to move in both the absolute and MINE/YOURS frames. The engine is differential-tested against the upstream reference of Li et al. (2023): byte-exact agreement over 500 games / 29,991 plies / 226 pass-plies, in both frames. It is the label oracle for every probe and the target for every intervention.

Model and data. Both models share a GPT-2-style backbone: 8 layers, width 512, 8 heads, learned absolute positions, full LayerNorm, exact-GELU, untied head (25.31M parameters). The AR control is causal (vocabulary 61: 60 move tokens + pad); the MDLM is the *same* backbone run bidirectionally with an added absorbing [MASK] token (vocabulary 62), trained with the loglinear masked-diffusion NELBO of Sahoo et al. (2024). Both are trained on the *same* 20M uniformly-random-legal games (the protocol of Li et al., 2023), so the only difference is the objective (every hyperparameter is listed in Appendix A). Both play Othello, and at near-matched competence: on teacher-forced next-move legality over 500 held-out games the AR scores 99.8% and the MDLM 98.0%; on free-generation whole-game legality, 99.4% against 90%. The per-move metric is the like-for-like comparator (whole-game legality compounds per-move errors over ~ 60 plies: $0.98^{60} \approx 0.30$, so the MDLM’s 90% is in fact better than naive compounding predicts). The 1.9-point per-move competence gap is far exceeded by the 17-point representational gap (0.79 vs. 0.96, §4) and the tenfold causal gap (+0.02 vs. +0.21, §5) it would have to explain, and §6 shows that, although it widens to 4 points on the hardest positions, it does not track the representational gap, so the weaker and barely-used board is attributable to the training objective, not to the MDLM being a worse Othello player.

Probes. Following Nanda et al. (2023), our headline metric `rel_2w` is a *linear* two-way probe predicting *mine* vs. *yours* on *occupied* cells, in the MINE/YOURS frame; `abs_2w` is its absolute-frame counterpart. We also fit nonlinear (2-layer MLP) variants (`*_2wnl`). Probes are trained on residual-stream activations after each block (1,000 training games) and evaluated on 300 held-out games, every occupied cell at every ply (17,700 and 17,997 posi-

probe	AR control	MDLM
linear, absolute	0.53	0.53
nonlinear, absolute	0.96	0.78
linear, relative	0.96	0.79
nonlinear, relative	0.96	0.79
untrained / shuffled	0.57/0.51	0.57/0.51

Table 1: Two-way board accuracy at the best layer (AR L5, MDLM L6); the linear-absolute row is at chance. The Li–Nanda dissociation reproduces in both models, yet the MDLM’s linear player-relative world model is substantially weaker (0.79 vs. 0.96). Metric names: `rel_2w` (linear, relative), `abs_2w` (linear, absolute), `2wnl` (nonlinear). Every layer: Table 7.

tions, 601,800 and 620,808 occupied-cell decisions for the AR and the MDLM), so exact binomial 95% CIs are below ± 0.01 for every accuracy we report, with untrained-weight and shuffled-label baselines throughout. We sweep *all* layers rather than inheriting the original’s layer choice. Comparing the absolute against the player-relative basis, the linear against the nonlinear form, every layer, and (below) the input masking regime and read order constitutes a set of ablations that isolate which choices the reported world-model signal actually depends on.

4 The World Model Transfers, but Is Weaker

Table 1 reports the probe battery. On the AR control we reproduce the Li–Nanda picture precisely: the board is linearly decodable *only* in the player-relative frame (linear-absolute at chance 0.53), while a nonlinear probe rescues the absolute frame. Our headline `rel_2w` saturates at 0.964 (epochs 1–4: $0.952 \rightarrow 0.960 \rightarrow 0.963 \rightarrow 0.964$), ~ 3 points below the original’s > 0.99 . We eliminate probe-side explanations: matching Nanda’s exact metric *lowers* the number (ruling out metric inflation), and crucially the *nonlinear* probe also tops out at ~ 0.96 , so the gap is how much board information the residual *contains*, not how it is encoded. Stratifying by game phase locates the shortfall (Table 9): on early- and mid-game positions the AR’s `rel_2w` is 0.9995 and 0.991; it is the late game (0.936), where the board is densest, that pulls the all-ply average to 0.96. We conclude it is an honest model-fidelity difference between our faithful reimplement on our corpus and the released checkpoint, exactly the kind of careful reproduction this track values.

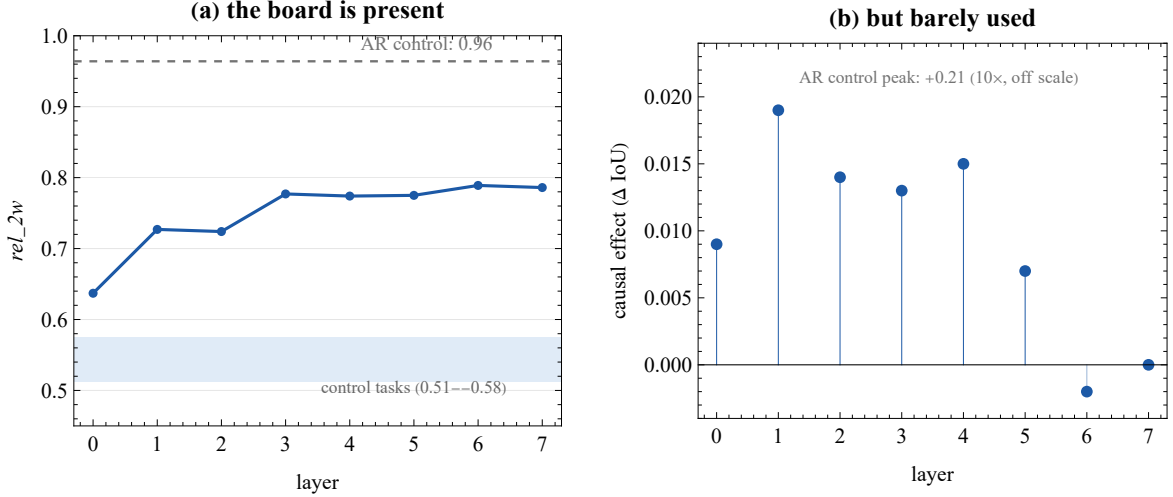


Figure 1: The masked-diffusion world model per layer, from the independent Rust reimplementaion (§8). **(a)** The player-relative linear probe (rel_2w) recovers the board at every depth, peaking at 0.79 (layer 6), far above the random-weight/shuffled control-task band but well below the AR control’s 0.96. **(b)** The causal effect of adding the probe direction is a floor ($\leq +0.02$); the AR control’s peak effect is $+0.21$, roughly ten times larger and off this scale. Present, but barely used.

Control tasks. Both models’ rel_2w sits far above two control tasks (Hewitt and Liang, 2019): a random-weight probe (~ 0.57) and a shuffled-label probe (~ 0.51). These bound the accuracy attributable to probe capacity or label memorisation rather than to the trained network, and place the 0.79 and 0.96 signals well outside the “plausible explanation from a random network” regime. Together with the latent (non-surface) nature of the board and the causal test of §5, they rule out that the probe is reading surface leakage rather than a used representation.

The same battery on the MDLM shows that the emergent linear world model *transfers*: the dissociation reproduces (linear-absolute at chance; linear-relative far above the 0.57/0.51 baselines; Figure 1a), and $rel_2w \approx 2wnl$ indicates the signal is information-limited, not nonlinearly hidden. But it is *markedly weaker*: 0.790 vs. the AR’s 0.964. This is a genuine objective-induced difference, not under-training: the MDLM’s free-generation legality and validation loss have plateaued, and its accuracy saturates ($0.731 \rightarrow 0.761 \rightarrow 0.776 \rightarrow 0.790$) toward an asymptote far below 0.96; indeed the AR after *one* epoch (0.952) already exceeds the MDLM after four. Probing under the model’s own forward masking only lowers decodability (the fully-revealed read is the peak), ruling out a denoising-regime artifact. To our knowledge this is the first emergent board-state world model probed in a

	effect	95% CI	win-rate
AR control (L2–3)	+0.21	± 0.017	84%
MDLM (L1)	+0.02	± 0.004	24%

Table 2: Causal intervention ($N=300$ AR, $N=2000$ MDLM). Adding the MINE/YOURS probe direction to flip a board cell steers the AR’s predicted moves strongly and consistently toward the counterfactual legal set (the causal claim of Nanda et al., 2023, reproduced), but moves the MDLM only minimally and inconsistently: the board is present yet barely used. Every layer: Table 8.

discrete-diffusion sequence model, and it shows the emergent linear world model is *objective-sensitive*.

5 The Board Is Present but Barely Used

A representation can be *present* without being *used*. Following Nanda et al. (2023), we test causal use: at a chosen occupied cell c whose colour flip changes the legal-move set, we add $\alpha \cdot \text{sign} \cdot \|h_t\| \cdot d_c / \|d_c\|$ to a block’s residual stream ($\alpha=1$ unless stated; swept in Table 3), where d_c is the *mine–yours* probe direction, and measure whether the model’s predicted moves shift toward the counterfactual legal set L' (intersection-over-union), versus a norm-matched random-direction control. We report the paired effect $\text{IoU}(\text{real}) - \text{IoU}(\text{random})$ with a 95% CI and a win-rate.

Table 2 gives the result. On the AR control the edit moves predictions toward the counterfac-

edit strength α	AR effect	MDLM effect
0.5	+0.22	+0.01
1.0	+0.21	+0.02
2.0	+0.25	+0.007
4.0	+0.49	+0.002

Table 3: Edit-strength (α) robustness of the best-layer causal effect ($N=2000$ for the MDLM, $N=300$ for the AR). The MDLM effect stays a floor ($\leq +0.02$) across the range, peaking at $\alpha=1$; the AR effect is strong at every α . “Barely used” is not an artifact of the chosen strength.

tual board by +0.21 at layers 2–3, in 84% of interventions (baseline IoU with the *true* legal set is 1.000: the AR plays legal Othello). Nanda’s causal claim reproduces on our pipeline. On the MDLM ($N=2000$) the effect is a *floor* (Figure 1b): +0.02, statistically real (the CI excludes zero), but $\sim 10\times$ smaller and, decisively, fired in only $\sim 24\%$ of interventions versus the AR’s $\sim 84\%$. The masked-diffusion model *represents* the board linearly yet barely relies on it to choose moves: a clean presence-vs-use dissociation against ground truth, and a small model’s honest floor. Notably the best *intervention* layers sit earlier than the best *probe* layer, which is why we sweep all layers.

This floor is robust to the edit strength. Sweeping $\alpha \in \{0.5, 1, 2, 4\}$ (Table 3), the MDLM’s best-layer effect never exceeds +0.02 and in fact peaks at $\alpha=1$, while the AR’s stays strong at every α (and grows as late-layer edits come to dominate). The presence-vs-use contrast is therefore not an artifact of the free parameter α .

6 Robustness: Position Difficulty and Seeds

Position difficulty. A 1.9-point gap in *average* legality could, in principle, hide a large functional gap on *hard* positions, which could then be what drives the weaker and barely-used board. We therefore stratify the three quantities of the argument (teacher-forced legality, `rel_2w` at the best probe layer, and the paired causal effect at the best intervention layer) by the branching factor $|L|$ of the position, the number of legal moves (the fewer, the harder a legal move is to find), and by game phase (Table 9). Table 4 gives the result. The competence gap does grow with difficulty, from 0.5 points on the easiest positions to 4 points on the hardest ($|L| \leq 3$, where the MDLM still plays 95% legal moves), but the representational gap does not track

	legality		rel_2w		causal effect	
$ L $	AR	MDLM	AR	MDLM	AR	MDLM
1–3	0.992	0.952	0.856	0.685	+0.25 [†]	+0.015 [†]
4–7	0.998	0.979	0.974	0.793	+0.21	+0.014
8+	1.000	0.985	0.989	0.825	+0.16	+0.011
all	0.998	0.980	0.964	0.790	+0.17	+0.012

Table 4: Difficulty stratification by branching factor $|L|$ (the number of legal moves at the position; fewer is harder). Teacher-forced next-move legality (500 held-out games), linear player-relative decodability at the best layer (300 held-out games), and the paired causal effect at the best intervention layer ($N=2000$ per model, edit plies spread uniformly over $[15, 50]$; Appendix A; spreading the plies lowers both models’ absolute effects relative to Table 2’s ply-15 edits, +0.166 / 75% and +0.012 / 18% overall, in the same $14\times$ ratio). [†]Only 24 interventions (± 0.13 and ± 0.06 CIs). The competence gap never exceeds 4 points, while the representational gap is 16–18 points in every stratum and the causal gap stays above tenfold: the latter are not a disguised competence gap.

it: it is 16–18 points in every $|L|$ stratum, and on the easiest stratum, which holds 60% of positions and where both models are at least 98.5% legal, it is 16.5 points. Both models’ decodability falls as the board fills (every stratum of Table 9), so the gap is smallest in the early game (7 points, where both models are at least 99.4% legal) and largest in the late game (20 points), but it never closes. The causal gap is $14\times$ overall, $15\times$ at $|L| \in [4, 7]$, $14\times$ at $|L| \geq 8$, and $10\text{--}18\times$ across game phases (the $|L| \leq 3$ bin holds only 24 interventions and is inconclusive on its own). Conditioning the probe read on the model’s *own* next-move legality is the sharpest version of the test: on the 98% of positions where the MDLM’s next move is legal, its board is decodable at 0.799; on the positions where the AR’s is legal, at 0.965. The representational gap is not concentrated where the MDLM misplays; it is the same everywhere.

Seeds. Every number so far comes from one training run per objective (seed 0). To bound run-to-run variance we trained two further seeds of each model with identical data, backbone, and budget (only the initialisation and the data order change) and re-ran the probe battery and the intervention on each. Table 5 reports the mean, standard deviation, and 95% confidence interval over the three runs. The AR’s linear board sits at 0.966 ± 0.002 and the MDLM’s at 0.794 ± 0.008 ; the best-layer causal effect at

	AR control	MDLM
next-move legality	0.9984 ± 0.0001	0.9784 ± 0.0011
free-gen. legality	0.994 ± 0.000	0.902 ± 0.003
re1_2w, best layer	0.966 ± 0.002	0.794 ± 0.008
causal effect, best layer	$+0.209 \pm 0.006$	$+0.016 \pm 0.004$
win-rate (%)	82.6 ± 3.1	23.6 ± 0.0

Table 5: Mean $\pm$ standard deviation over three training seeds per objective (seeds 0–2; every other table uses seed 0). The 95% confidence interval of each mean is $\pm 2.48 \text{ sd } (t_{0.975,2}/\sqrt{3})$. Legality over 500 held-out games (teacher-forced) and 200 (AR) / 128 (MDLM) generated games; re1_2w at each seed’s best layer (AR: 5, 5, 5; MDLM: 6, 4, 5: 0.964, 0.966, 0.967 vs. 0.790, 0.803, 0.787); causal effect and win-rate at each seed’s best intervention layer with $N=2000$ for both models (AR: layers 3, 4, 2: +0.215, +0.204, +0.209; MDLM: layer 1 in every seed: +0.020, +0.012, +0.015).

frac.	random	prefix	suffix	conf.	spread
0.25	0.728	0.740	0.740	0.710	0.030
0.50	0.671	0.668	0.659	0.638	0.033

Table 6: Order-invariance of probing (MDLM re1_2w at L6) at mask fraction *frac*. At a fixed fraction the probe is nearly invariant to *which* positions are masked (spread ≤ 0.033); the fully-revealed reference is 0.790. The model’s own confidence order (bold) reads *lowest*, the opposite of what a decode-order exploit would produce.

+0.209 $\pm$ 0.006 against +0.016 $\pm$ 0.004, the win-rate at 83% against 24% in every MDLM seed. The seed spread is an order of magnitude below the 17-point representational gap and the 13-fold causal gap, and the confidence intervals do not come close to touching: the attribution to the objective is not a single-seed accident. The MDLM’s best *probe* layer does move with the seed (6, 4, 5) while its best intervention layer stays at layer 1, and the AR’s best intervention layer moves (3, 4, 2) while its best probe layer stays at layer 5, which is why every table sweeps all layers. Measurement seeds matter even less: refitting every probe with three seeds on the fixed seed-0 checkpoints moves re1_2w by at most 0.001 at any layer (Table 7), and re-extracting the steering directions and redrawing the random control keeps the best-layer effect within +0.204 to +0.207 (AR) and +0.017 to +0.020 (MDLM) with win-rates within one point (Table 8).

7 Ruling Out a Decode-Order Artifact

Because an MDLM can unmask in any order, a world-model claim must exclude the possibility that the apparent representation is an artifact of an

easy decoding order (Trainin et al., 2026). Two observations rule this out. First, *by construction*: our headline 0.79 is measured on *fully-revealed* games ($t=0$), which contain no masked slots and therefore no unmasking order to exploit. Second, we run positive controls. **(i) Probe order-invariance.** At a fixed mask fraction we vary *which* positions are masked (random, a leading block, a trailing block, or the model’s own most-confident-first order) and re-probe at the fixed best layer. Accuracy is nearly invariant (spread ≤ 0.033 across patterns at each fraction; Table 6), and the model’s *own* confidence order reads *lowest*, the opposite of what an order-exploit would produce. **(ii) Causal use in the model’s own order.** Reading the intervention at the model’s own next-decode position rather than the game-order position leaves the effect at the floor (+0.012, win-rate 17%, no larger than the game-order read), and the two orders agree 64% of the time. The weak, barely-used world model is genuine, not a decode-order trick.

8 An Independent Reimplementation

Reproducibility is the point of this track, so beyond releasing our (Python) code we reproduced the headline results a second time in an *independent* implementation: diakrisis, a Rust engine and probing/intervention harness built on candle-mi (Jacopin, 2026a), an open-source mechanistic-interpretability library for the Rust candle framework, sharing no code with the primary (PyTorch) pipeline. The independent forward pass matches the reference logits and every per-layer residual stream to a worst-case 2.6×10^{-4} (the exact tensors the probes consume). The Rust port of the engine is byte-exact against the differential-tested reference over 6,000 plies. On this stack the linear probe reproduces re1_2w = 0.789 at the best layer (vs. 0.790), and the causal intervention reproduces the floor (+0.019, win-rate 25% at L1, vs. +0.020 / 24%). Two independent implementations agreeing to $\sim 10^{-3}$ rule out a reference-side numerical or wiring artifact behind any of our claims. Both implementations, the data recipes, the released fp32 checkpoints, and per-table run commands are available in the public repository (Jacopin, 2026b).

9 Conclusion

Holding backbone and data fixed and changing only the objective, we find that the emergent linear Othello world model *transfers* from autoregres-

sive to masked-diffusion training but is *substantially weaker* (0.79 vs. 0.96) and only *minimally used* (+0.02 / 24%): a presence-vs-use dissociation against ground truth, cleared of the decode-order confound, stable across position difficulty and training seeds, and reproduced in a fully independent implementation. The emergent linear world model is objective-sensitive: masked diffusion yields a measurably less linearly-crisp, and far less causally load-bearing, internal board than autoregression on the same backbone and data.

Limitations

Our MDLM is small (25M parameters) and we study a single backbone size; we frame its causal effect explicitly as a *floor* and do not claim it bounds larger masked-diffusion models. Each objective is represented by three training runs of one configuration (§6); the seed spread they bound is that of this backbone, data, and budget, not of other sizes, hyperparameters, or corpora. We study a single synthetic task (Othello); the transfer of these findings to natural language is open. We do not run the bespoke “asymmetry-probe” positions of Trainin et al. (2026); given a world model already shown to be weak, barely causal, and order-invariant, we judge it low-value and leave it to future work. Finally, we study *presence* and *use* of the board, not the temporal question of *when* across denoising steps a masked-diffusion model commits to a move; that commitment-timing question is the natural next study.

References

- Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.
- Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, pages 17981–17993. ArXiv:2107.03006.
- Yonatan Belinkov. 2022. [Probing classifiers: Promises, shortcomings, and advances](#). *Computational Linguistics*, 48(1):207–219.
- Gerard Boxo, Aman Neelappa, and Shivam Raval. 2025. Linear probes rely on textual evidence: Results from leakage mitigation studies in language models. *arXiv preprint arXiv:2509.21344*.
- Dean S. Hazineh, Zechen Zhang, and Jeffery Chiu. 2023. Linear latent world models in simple transformers: A case study on Othello-GPT. *arXiv preprint arXiv:2310.07582*.
- John Hewitt and Percy Liang. 2019. [Designing and interpreting probes with control tasks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.
- Éric Jacopin. 2026a. [candle-mi: Mechanistic interpretability for the rust of us](#). Rust crate, version 0.1.24.
- Éric Jacopin. 2026b. [Code, data, and model checkpoints for “present but barely used”](#). GitHub repository. Both implementations, the data recipes, and the released fp32 checkpoints.
- Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. [Emergent world representations: Exploring a sequence model trained on a synthetic task](#). In *The Eleventh International Conference on Learning Representations (ICLR)*. ArXiv:2210.13382.
- Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 32819–32848. PMLR. ArXiv:2310.16834.
- Maxime Méloux, Giada Dirupo, François Portet, and Maxime Peyrard. 2025. The dead salmons of AI interpretability. *arXiv preprint arXiv:2512.18792*.
- Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. [Emergent linear representations in world models of self-supervised sequence models](#). In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 16–30, Singapore. Association for Computational Linguistics.
- Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander M. Rush, and Volodymyr Kuleshov. 2024. [Simple and effective masked diffusion language models](#). In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*. ArXiv:2406.07524.
- Shubham Toshniwal, Sam Wiseman, Karen Livescu, and Kevin Gimpel. 2022. [Chess as a testbed for language model state tracking](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11385–11393. ArXiv:2102.13249.
- Itamar Trainin, Shauli Ravfogel, Omri Abend, and Amir Feder. 2026. Discrete diffusion models exploit asymmetry to solve lookahead planning tasks. *arXiv preprint arXiv:2602.19980*.

Yifei Yuan and Anders Søgaard. 2025. Revisiting the Othello world model hypothesis. *arXiv preprint arXiv:2503.04421*.

A Experimental Details

Everything below is reproducible from the public repository (Jacopin, 2026b), whose README maps each table to a command; the values here are the ones those commands use.

Engine and corpus. The rule engine is a dependency-free Python reimplement of the upstream rules (implicit passes, game end when neither side can move). Games are generated with the uniformly-random-legal protocol of Li et al. (2023): 20,000,000 training games (generator seeds 0..20,000,000 - 1) and 100,000 validation games (seeds $10^9..10^9+99,999$), so the two splits are disjoint by construction. Each game is a sequence of at most 60 move tokens (1..60, the 60 playable cells in ascending order), right-padded with token 0 to length 60; the MDLM adds [MASK] = 61. All probe, intervention, and competence evaluations draw from the validation split: probe training uses its first 1,000 games, probe testing the next 300, teacher-forced legality its first 500 games, and interventions are drawn from its start until N are collected.

Backbone. A GPT-2-style transformer: 8 pre-LayerNorm blocks, $d=512$, 8 heads of dimension 64, MLP width $4d$ with the exact (erf) GELU, learned absolute positional embeddings, LayerNorm with bias at every site, an untied output head without bias, no dropout, PyTorch default initialization; 25.31M parameters. The AR control uses a causal attention mask, vocabulary 61, and a 59-token context (it predicts token $i+1$ from tokens 0.. i); the MDLM removes the mask (bidirectional attention) and uses vocabulary 62 over the full 60-token sequence. Nothing else differs.

Training. AdamW (PyTorch defaults: $\beta=(0.9, 0.999)$, weight decay 0.01), constant learning rate 3×10^{-4} , no warm-up, no schedule, no gradient clipping; batch 512 games; 4 epochs over the 20M games (39,062 steps per epoch, 156,250 in total); bf16 autocast with torch.compile; seeds 0, 1, 2 (the per-epoch shuffle is seeded by seed + epoch; seed 0 is the run every table but Table 5 uses). AR objective: next-token cross-entropy, pad ignored. MDLM objective: the loglinear absorbing-state NELBO

of Sahoo et al. (2024): per sequence, a noise level $t \sim U[\epsilon, 1)$ with $\epsilon=10^{-3}$, each non-pad token masked independently with probability t , cross-entropy on the masked tokens weighted by $1/t$; pad tokens are never masked nor scored. Validation every 500 steps: held-out legal-move accuracy on 200 games (AR); mean diffusion loss on 4 batches and free-generation legality of 128 games sampled with a 64-step ancestral denoiser (MDLM). Throughput on one RTX 5060 Ti (16 GB): $\sim 188k$ tokens/s for the AR (~ 1.75 h per epoch, 7 h in total) and $\sim 177k$ tokens/s for the MDLM (~ 1.9 h per epoch, 7.5 h in total). Software: PyTorch 2.10 (CUDA 13), Python 3.14. The seed-0 MDLM’s epoch 1 was trained in one session and epochs 2–4 resumed from its checkpoint (weights, optimizer moments, step); the seeded shuffle makes this equivalent to an uninterrupted run, and the seed-1 and seed-2 runs were uninterrupted. The seed replicas were run by scripts/seed_replicas.sh in the repository.

Probes. Activations are the residual stream after each of the 8 blocks, at every ply of a fully-revealed game; the label at ply i is the board after moves 0.. i , from the engine, in the absolute (black/white/empty) and player-relative (MINE/YOURS/empty) frames. 2w probes are a single linear map $512 \rightarrow 64 \times 2$ trained with cross-entropy on occupied cells only (empty cells are ignored); 2wnl probes are $512 \rightarrow 128 \rightarrow 64 \times 2$ with a ReLU; the 3-class cell probe (not reported in the paper) is $512 \rightarrow 64 \times 3$ over all cells. All probes: AdamW, learning rate 10^{-3} , 50 epochs, batch 1,024 positions, on 1,000 games (58,956 AR / 59,938 MDLM positions), evaluated on 300 held-out games (17,700 AR / 17,997 MDLM positions, i.e. plies 0..58 and 0..59 respectively; 601,800 and 620,808 occupied-cell decisions). Accuracy is the fraction of occupied cells whose colour is predicted correctly. Control tasks (Hewitt and Liang, 2019): the *untrained* control fits the same probe on the same architecture with freshly-initialised weights; the *shuffled* control permutes the labels across positions on the trained model’s activations. Probe seeds: the reported numbers use seed 0 (probe initialisation and minibatch order seeded as seed + layer); Table 7 reports seeds 0..2.

Masking regimes. The order controls of §7 corrupt the input before the forward pass while keeping the clean latent board as the label. Stochastic masking (-mask-t) masks every move indepen-

dently with probability t ; fixed-fraction patterns mask exactly $\lfloor f \cdot L \rfloor$ of a game’s L positions: a random subset (seeded), the trailing block (*prefix* revealed), the leading block (*suffix* revealed), or the $\lfloor f \cdot L \rfloor$ positions the model is least confident about, ranked by the top move-token probability of one fully-masked forward (*confidence*: the model’s own order).

Causal intervention. Steering directions come from a seeded linear `rel_2w` probe at every layer (seed 0 + layer, same budget as above): $d_c = W_{c,\text{MINE}} - W_{c,\text{YOURS}}$ for each of the 64 cells. Interventions are collected from validation games at plies $t \in [15, 50]$, at most 2 per game, over occupied cells c whose colour flip changes the legal-move set ($L' \neq L$, both non-empty); the collector takes the first two qualifying (t, c) pairs of each game, which in practice places nearly every edit at ply 15–16 (the stratified Table 4 instead draws two plies per game uniformly from $[15, 50]$ and one qualifying cell per ply at random). The edit adds $\alpha \cdot \text{sign} \cdot \|h_t\| \cdot d_c / \|d_c\|$ to the residual stream after block ℓ at the last revealed position t , with $\text{sign} = -1$ for a MINE disc and $+1$ for a YOURS disc, and $\alpha=1$ (so the edit has the norm of the residual it is added to). The AR is read at position t ; the MDLM reveals the prefix $0..t$, masks the rest, and is read at the masked slot $t+1$ (game order) or at its most-confident masked slot (own order). The prediction is the top- $|L'|$ move tokens (pad and [MASK] excluded); the score is IoU with L' . The control repeats the edit with a Gaussian random direction of the same norm (seeded). The effect is the mean paired difference with a 1.96 SE CI; the win-rate is the fraction of strictly positive differences. $N=300$ (AR) and $N=2000$ (MDLM) in the paper’s Tables 2 and 3; $N=2000$ for both in Tables 4 and 8.

Competence. Teacher-forced next-move legality: for every ply i of 500 validation games (29,490 scorable positions), the model’s argmax next move given the true prefix $0..i$ (AR: one causal forward per game; MDLM: prefix revealed, rest masked, read at $i+1$) is checked against the legal set after moves $0..i$; the submitted version’s 99.9% for the AR was the training-time evaluation of the same metric on 200 games. Free-generation legality: whole games are generated (AR: sampling at temperature 1 from a uniformly-random legal opening, 200 games; MDLM: 128 games from the 64-step ancestral denoiser at temperature 1, the training-

time evaluation) and replayed on the engine; the score is the fraction of attempted moves that are legal, replay stopping at the first illegal move.

Code. The repository is a single Python package, `othello_md1m`, with no dependency beyond PyTorch, NumPy, and safetensors: `engine/` (rules, oracle, tokenizer, seeded game generation), `data/` (corpus generation and memory-mapped loading), `models/` (the shared backbone as `OthelloGPT` and `OthelloMDLM`, differing only in the causal flag and the vocabulary), `train/` (the two training loops and the diffusion objective), `probe/` (activation capture, the masking patterns, the probes and the control tasks), `intervention.py` and `run_intervention.py` (direction extraction and the causal test), `eval_competence.py` and `eval_by_difficulty.py` (§3 and §6), and the two fixture exporters consumed by the Rust crate. Every pure helper carries a doctest; every script is seeded and takes its budget on the command line; the README maps each table of this paper to one command, and the released checkpoints let every table but the training curves be regenerated without retraining.

Independent reimplementations. The Rust crate `diakrisis` (engine, probe, and the `elenchos` intervention) is built on `candle-mi` (Jacopin, 2026a) (v0.1.18 at the time of the parity run; its `OthelloGpt` backbone loads the released state dict verbatim). Forward parity against the PyTorch reference on exported fixtures: logits to 4.2×10^{-5} , every per-layer residual stream to 2.6×10^{-4} (CPU, fp32). The Rust engine is byte-exact against the Python engine on 100 traced games (6,000 plies: mover, board, side to move, legal set after every ply). Its probe uses the same recipe (linear $512 \rightarrow 64 \times 2$, AdamW 10^{-3} , 50 epochs, batch 1,024, 1,000 train / 300 test games generated from seeds by the Rust engine) and its intervention the same protocol, over all 8 layers.

B Per-Layer and Per-Stratum Tables

The body of the paper reports each quantity at its best layer or as an average over positions; the three tables here give the full sweeps behind them (seed-0 checkpoints). Table 7 is the probe battery of §4 at every layer of both models, with the control tasks and the spread over three probe seeds. Three things the best-layer summary hides: the linear-absolute probe is at chance at *every* layer of both models

layer	AR control				MDLM			
	abs_2w	abs_2wnl	rel_2w	rel_2wnl	abs_2w	abs_2wnl	rel_2w	rel_2wnl
0	0.537	0.742	0.755	0.753	0.539	0.625	0.636	0.636
1	0.535	0.814	0.823	0.819	0.537	0.714	0.727	0.728
2	0.533	0.892	0.897	0.895	0.535	0.711	0.723	0.725
3	0.533	0.931	0.935	0.932	0.535	0.766	0.777	0.776
4	0.530	0.950	0.953	0.951	0.533	0.765	0.777	0.774
5	0.529	0.960	0.964	0.962	0.531	0.765	0.776	0.775
6	0.527	0.859	0.957	0.957	0.531	0.778	0.790	0.789
7	0.525	0.854	0.954	0.956	0.532	0.772	0.786	0.785
untrained			0.576					0.574
shuffled			0.501					0.514