n-CLU: A FACTORIZED LATENT DYNAMICS MODEL FOR DELETABLE COMPOSITIONAL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A world model trusted with unseen tasks needs to go beyond generalizable predictions. Among other capabilities, dynamic tasks demand that (a) the model outputs are *valid* under the task's judge (b) the model *composes* new combinations of factors from the ones it has seen, and (c) the model can *forget* specific factors when the environment changes permanently. Valid, compositional generation is the primary focus of this work with machine unlearning as a supplementary goal. We present the *naive-Causal Learning Unit* (n-CLU), and compare against naive baselines (an MLP, an LSTM, a transformer and a continuous-input MLP) on two simulated evaluation tasks: shape-rendering and maze-solving. The n-CLU is designed as a set of factorized, principal subspaces acting on the latent space of a frozen neural-network encoder, allowing composition of encoder-extracted features. The n-CLU is then read by a request-indexed address book, and finally rendered by a frozen decoder. Deleting a training row only downdates the statistic and refits the address book, so the trained n-CLU after deletion equals an n-CLU fitted without those rows. On a held-out set of the Shapes3D dataset, the n-CLU composes new pairs and combinations of every factor. Against matched baselines trained on the same frozen encoder and decoder, the n-CLU produces ~ 7 more distinct, valid novel tuples per held-out combination than the strongest baseline, and beats every baseline on every evaluation set of an n-way combination task across multiple seeds. To test compositional generation in a harder, world-model aligned task, we created a maze-based benchmark (CMAZE-T) with composable factors and validity rules an agent must follow during navigation, based on the state of the local environment. We also release this small-scale benchmark for valid, compositional generation and plan to expand on it in future works. On CMAZE-T the n-CLU consistently matches the strongest baseline (transformer) with a marginal seed-dependent lead, and we show both quantitative and qualitative results for the same. We finally motivate a Physics-designed extension to the n-CLU, and how it could provide clues towards designing better latent-dynamics based world models.

1 INTRODUCTION

Generating held-out combinations of factors is an active problem (Du et al., 2020; Liu et al., 2022) in the AI community. Methods targeting composition, like the multilinear class of algorithms (Tenenbaum & Freeman, 2000, TensorFaces (Vasilescu & Terzopoulos, 2002), Vlasic et al., 2005, MLC-VAE (Georgopoulos et al., 2020)) compose by factorizing a data tensor organized by every label combination and rendering linearly. Other approaches based on the frozen-generator method (GANSpace (Härkönen et al., 2020), TCA (Oldfield et al., 2021), PandA (Oldfield et al., 2023) etc.) find editable directions in a pretrained generator without labels and without an address book.

On the other hand machine unlearning, that matches a retrain without the deleted rows, is also an active research area and a useful feature to have in a world model. K-means (Ginart et al., 2019) based methods, bilinear matrix completion (Xu et al., 2023), and ridge heads on frozen backbones (Quan et al., 2027) are common approaches, while principal-component eigen decomposition and generative models are relatively less explored. Recent generative-model unlearning attempts are approximate (Kong & Alfeld, 2022; Kong & Chaudhuri, 2023; Moon et al., 2024; Sun et al., 2025) applied to pretrained GANs and VAEs unlearn extracted features rather than input rows, and are ap-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

proximate by construction. SISA training (Bourtoule et al., 2021) and compartmentalized diffusion models (Golatkar et al., 2024) on the other hand rely on retraining shards of a model and training multiple copies respectively and dropping the one that trained on the row to be deleted. To the best of our knowledge, the literature doesn't contain a unified solution for generation and input-row deletion.

Therefore, we present the naive-Causal Learning Unit (n-CLU), that combines multilinear factored generation, a request-indexed tensor head, and the downdate primitive, which follows from the summation form of Cao & Yang (2015) into a single design acting as a Latent Dynamics Model (LDM). In this work we define an LDM as an algorithm that is either designed or learns to, enforce or extract the dynamical rules of a latent space parametrized by another independent learning algorithm. We later show how our n-CLU is a trivialization of a Physics-structured CLU, in-turn inspired by the CHLU presented in Jawahar & Pierini (2026). We believe LDMs like the CLU, as the name suggests, can provide actionable clues towards building AI models that enrich and leverage latent spaces better.

1.1 World model properties

A world model (Ha & Schmidhuber, 2018; Hafner et al., 2025), among other capabilities, must be able to generate valid, unseen states and transitions (Valevski et al., 2025). An output is valid if the task's judge (regardless of whether a ground-truth answer exists) accepts it (Vafa et al., 2024) and approximate similarity to a training item does not count. Composition here, is defined as the model producing valid outputs for held-out combinations of factors it has seen only separately (Zhao et al., 2022) or in a different composition. Deletion another useful feature and is defined as the algorithm matching a copy of itself that was never trained on an input or a factor, after that input or factor is deleted. The three pull against each other: a lookup table can delete in this sense but cannot compose anything not already stored; a learned generator composes but generally achieves deletion only by retraining (the whole model, or a shard of it (Bourtoule et al., 2021)); and approximate unlearning (Guo et al., 2023; Golatkar et al., 2020) is non-exact by construction. The n-CLU is a unified approach that combines some of these properties as explained below.

1.2 Positioning and scope

The n-CLU is a primitive unit: one model with no harness built around it, in the sense in which MLPs, LSTMs (Hochreiter & Schmidhuber, 1997) or a transformers (Vaswani et al., 2023) are AI primitives. We compare it only with other matched, naive units, an MLP, an LSTM, a transformer and a continuous-input MLP (Sec. 2). Systems built on top of units, such as sharded retraining, compositional diffusion and physics-informed variants of an MLP, are out of scope by definition.

1.3 Contributions

1. Compositional generation, against matched baselines on Shapes3D (Burgess & Kim, 2018; Kim & Mnih, 2019) and a world models-aligned maze dataset, CMAZE-T.
2. Deletion on the n-CLU and after a value, conjunction or item. Post-deletion, the n-CLU matches an n-CLU that was never trained on the deleted rows. However, we do not make a general claim about compositional refusal after input row deletion.
3. A small-scale composition and deletion benchmark task (CMAZE-T) as a minor contribution. The benchmark consists of randomly generated maze grids, with colored doors, colored keys, tinted floors and a set of valid compositional rules that make maze solving less trivial and demand the algorithm to learn compositions to predict valid futures.

2 Methods

Jawahar & Pierini (2026) introduced the CHLU as an LDM. We build on top of this idea and drop the physics-based design to construct our naive-CLU as a special case of the CHLU and test it as a factorized memory unit for compositional generation with additional properties such as deletion as a virtue of its design.

2.1 ARCHITECTURE AND SETUP

The pre-requisites for the n-CLU as defined here are a pre-trained, frozen encoder that maps each training item to a latent space (this latent vector serves as the n-CLU input) and a pre-trained frozen decoder or downstream task head (takes n-CLU outputs and produces the output structure the task demands). The latent space is partitioned, without labels, into K regions called wells by a balanced correlation partition. In each well the n-CLU keeps a fixed number of atoms, a hyperparameter: the mean of the well's training latents and their leading principal directions. The atoms are not stored as such but derived from a retained per-well statistic, the count, sum and second-moment matrix of those latents, so that deleting rows amounts to subtracting them from the statistic and re-deriving the atoms. We define n to be the number of factors composed for a given request at inference. Details of how $K^\star$ is chosen are explained and tabulated in App. A, Tab. 4).

The read step: An inference request is a tuple of factor values. n-CLU contains a tensor-product address book, defined over the training set, that maps the request to per-well coefficients. The read step renders $x_k = U_k c_k$ in each well's orthonormal basis and the frozen decoder renders the image. The read itself is request-only and no feature tensor enters it. Labels therefore touch the model only at the address book. This is the interface split: unsupervised context from a frozen encoder, a request-indexed head, and a frozen renderer, with no Cartesian-product training requirement (Wiedemer et al., 2023; Lachapelle et al., 2023; Montero et al., 2022).

The memory footprint: Every n-CLU row shows four byte counts separately: (1) the atoms, (2) the covariance and count statistic, (3) the address book, and (4) a coupling and hole table used for refusal. The 524,288-byte figure that recurs below is an arbitrary reference choice and the matched baselines carry about 131,072 (float32) learnable parameters.

Factor relations under the hold-out draw: The hold-out removes factor combinations from the training set at random. On some draws, for some factors, it removes every training instance. A model that encodes, say color, by identity (like a lookup table) then has no training evidence for the held-out instances of that color and cannot answer them, however it is trained. A model that is also given the relation "wall color equals tint color" can: the relation is learned from the other colors, where it is seen, and applies unchanged to the color whose rows were removed. This is what the relational columns add (Fig. 4(c)), and it is why every seed is reported separately for relational and non-relational models.

The judge: On Shapes3D an external classifier, judges each rendered image; the metric is the number of *distinct valid novel factor tuples per held-out combination* (Tab. 1). The results are reported over combinations with a margin of 1.0 tuple per held-out combination.

The retained statistic and deletion: Along with the context the n-CLU also keeps per well, the running covariance, mean and count of the fit rows. Deleting rows subtracts their contribution from the statistic. The atoms are re-eigen decomposed from the subtracted (downdated) statistic. The address book keeps its own retained statistic, the count and first moment of its fit rows; deleting rows downdates it, and the book is refit from it.

3 EXPERIMENTS

The two datasets, Shapes3D and CMAZE-T are visualized Figs.1 and 2 respectively, and their setups are explained in the captions.

Baselines: Learned baselines (MLP, LSTM (Hochreiter & Schmidhuber, 1997), transformer (Vaswani et al., 2023), and a continuous-input MLP) are matched to the n-CLU's atoms, and fitted from the same frozen features. At $K^\star$ the fitted baselines carry the parameter counts of Tab. 1 and the n-CLU incurs the additional address book count. The table shows one capacity matched diagnostic (continuous MLP at 610k params, which loses to the CLU on all seeds). A lookup table of written items is a non-learned comparison floor in our case. The four controls (landscape controls) that help isolate the contribution of the n-CLU over the frozen encoder-decoder are:

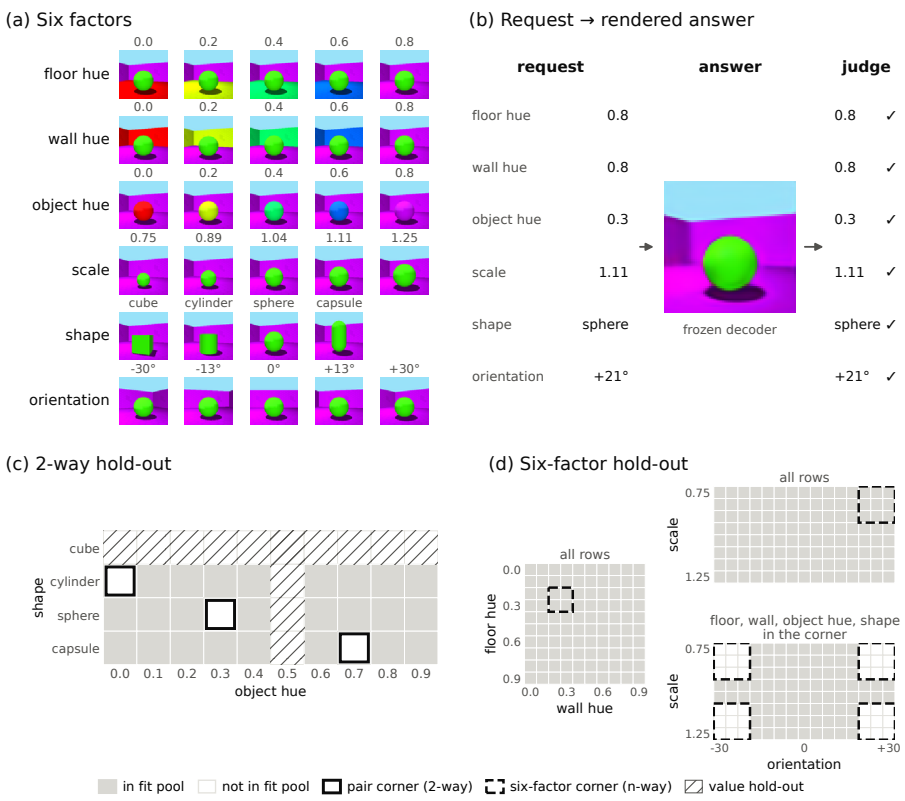


Figure 1: The Shapes3D task: (a) The six factors (floor hue, wall hue, object hue, scale, shape, orientation), each row varying one factor of real dataset images with the other five held at the request's values. (b) A request is a tuple of factor values; the answer is an image rendered by the one frozen decoder, and the judge is the external classifier, which must read back all six factors. (c) The two-way hold-out on object hue × shape (d) A six-factor combination hold-out example.

landscape removed (atoms zeroed), *landscape shuffled* (atom values permuted within each well), *shuffled across wells* (well blocks dealt to other wells) and *random atoms* (random atoms in place of the fitted ones).

n-way composition: This experiment checks the n-CLU's composition performance as more factors are requested jointly in a single inference pass out of different hold outs. Tab. 2 shows the n-CLU at $K^\star$ on every n and seed beside matched baseline sets on the same combinations. The landscape controls at $K^\star$ collapse on every seed as seen in Tab. 1, showing the n-CLU is not being by-passed by the encoder-decoder.

Deletion: Three deletions are run at $K^\star$, on its fit pool (Tab. 6), at three seeds. *Value*: every fit row carrying red. *Conjunction*: every fit row carrying red *and* sphere. *Item*: a random 1% of fit rows, which is also the control for refit instability. *Forgetting* c is three operations: downdate the n-CLU's context; downdate the address book's statistic and refit the book from it; and punch the hole for c (Sec. B.2) for post-deletion refusal.

4 Results

4.1 Composition on Shapes3D

Tab. 1 shows the full composition results on Shapes3D. Row 3 holds out 60 pair combinations and row 4 holds out 49 six-factor combinations; the n-CLU is compared with each baseline on all 109 combinations.

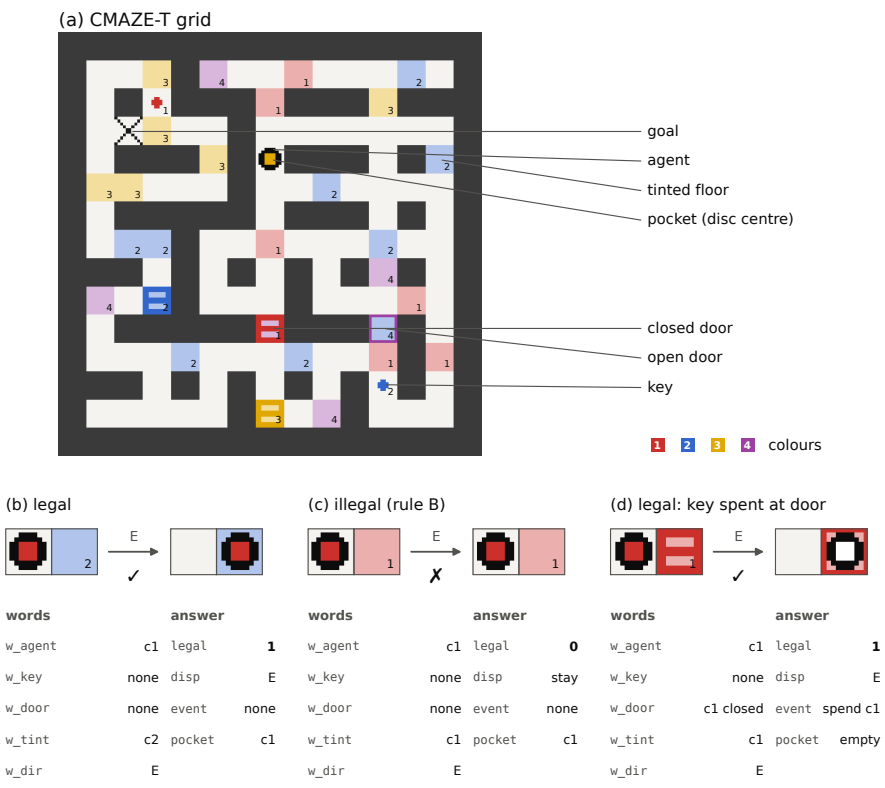


Figure 2: The CMAZE-T task: (a) One generated maze, drawn by the engine: walls, floor tinted in one of four colors, colored keys (diamonds), closed and opened doors, the goal (a cross), and the agent: a black disc whose center shows the key in its pocket, white when the pocket is empty (it holds at most one key). Rule A: a closed door of color c needs key c to open and consumes the key after. Rule B: stepping on floor tinted c is forbidden while holding key c. Rule A is applied before rule B. (b)–(d) One transition each: the inference request is five words (`w_agent`, `w_key`, `w_door`, `w_tint`, `w_dir`) and the answer is four facts (`legal`, `disp`, `event`, `pocket`), computed by the engine and checked against an independent stepper. (b) A legal move onto floor of another tint. (c) An illegal move under rule B: the agent does not move, so after equals before and the verdict is carried by the mark and by `legal` $= 0$. (d) The same pocket and tint as (c) with a closed door of the pocket's color: the door consumes the key first, so the move is legal. Panels (c) and (d) differ only in the door, which is the pocket $\times$ door $\times$ tint interaction held out in this example.

Qualitative results: Figs. 3 and 8 show single held-out requests from Tab. 1 as images, at seed 0 and $K^\star = 32$, against four baselines the lookup table. Every model's output on an inference request is drawn through the same frozen decoder, and each panel carries the judge's verdict on that image. The CLU gets most combinations right when atleast one baseline fails. When the n-CLU is incorrect, its image is often wrong in one factor only (most often the scale or the orientation, less often the shape), which is potentially a systematic failure mode at the current configuration. Requests on which a baseline succeeds and the n-CLU fails exist for every baseline and are not confined to the capsule (Fig. 8).

4.2 THE n-WAY COMPOSITION

n-CLU outperforms all baselines at every n as seen in Tab. 2. However, the rise from 2-way to 6-way is not to be inferred, because rungs 3–6 probe overlapping combinations and differ strictly only in fit-pool coverage. What varies is how coarsely the hold-out set is carved, and with it the held-out row count. Regardless, on every rung and every seed the n-CLU's score in Tab. 2 exceeds every baseline's score (Fig. 7).

Table 1: Composition at $K^\star = 32$ against matched baselines on Shapes3D, 3 seeds (0, 1, 2). Every baseline (no. of training steps in brackets) is matched to the n-CLU's atoms at $\sim 130k$ parameters (MLP diagnostic at $610k$ params.). "Paired Δ" is the n-CLU minus that baseline over all 109 combinations. LACE (Nie et al., 2021) and PoE (Hinton, 2002) are in Tab. 3. The n-CLU's row-3 and row-4 means are 32.350 ± 0.815 and 67.177 ± 1.320 ($\pm$ the sample standard deviation over the three seeds).

arm (optimiser steps)	seeds	all 109 combinations seed 0 / 1 / 2	(60 pairs held out) 3-seed mean	(49 $n = 6$ held out) 3-seed mean	paired Δ, n-CLU $-$ arm seed 0 / 1 / 2	combinations won / tied / lost seed 0 · 1 · 2
n-CLU, $K^\star = 32$ (closed form)	3	**47.495 / 47.404 / 49.119**	**32.350**	**67.177**	—	—
MLP (20,480)	3	36.706 / 37.569 / 43.174	25.394	55.993		
MLP (204,800)*	3	40.009 / 38.807 / 44.101	26.417	58.796	+7.486 / +8.596 / +5.018	76/11/22 · 84/9/16 · 69/27/13
MLP (614,400; diagnostic)	3	41.477 / 36.541 / 43.495	25.006	59.483	+6.018 (seed 0)	67/14/28 (seed 0)
LSTM (20,480)*	3	33.972 / 32.358 / 40.697	23.467	50.626	+13.523 / +15.046 / +8.422	
LSTM (204,800)	3	33.688 / 32.541 / 39.853	22.994	50.503		
transformer (20,480)*	3	25.560 / 27.789 / 38.651	20.050	43.667	+21.936 / +19.615 / +10.468	
transformer (204,800)	3	25.303 / 27.239 / 37.844	18.783	44.020		
continuous MLP (20,480)	3	17.917 / 16.541 / 26.844	11.722	31.102		
continuous MLP (204,800)*	3	28.532 / 32.936 / 42.596	22.589	49.503	+18.963 / +14.468 / +6.523	
continuous MLP (614,400; diagnostic)	3	27.284 / 34.009 / 38.706	21.894	47.340		
lookup table (floor)	3	8.229 / 5.073 / 9.752	5.167	10.769		
four landscape controls	3	0.000 (all)	0.000 (all)	0.000 (all)		

Table 2: n-way composition on Shapes3D, 3 seeds. Distinct valid novel tuples per combination reported, per seed (0 / 1 / 2). n-CLU outperforms all baselines at all combination-seed pairs.

arm	seeds	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$
n-CLU, $K^\star$	3	**31.800 / 32.500 / 33.633**	**60.150 / 62.783 / 61.908**	**65.242 / 65.817 / 67.200**	**65.642 / 66.817 / 67.283**	**65.908 / 66.250 / 67.650**
3-seed mean	3	32.644	61.614	66.086	66.581	66.603
data-matched set: each rung's own 131,072 rows; 20,480 steps, and 204,800 for the MLP and the continuous MLP						
MLP (20,480)	3	22.450 / 25.200 / 29.267	43.558 / 46.742 / 51.592	45.408 / 48.642 / 57.300	51.192 / 52.825 / 58.842	49.525 / 54.692 / 58.867
MLP (204,800)	3	27.167 / 25.583 / 29.783	47.258 / 46.833 / 57.200	50.517 / 50.017 / 60.075	53.642 / 53.275 / 60.767	54.183 / 52.200 / 60.033
LSTM (20,480)	3	20.550 / 22.383 / 27.150	36.400 / 37.233 / 40.908	42.492 / 39.958 / 53.517	46.492 / 47.442 / 55.692	47.758 / 46.275 / 55.417
transformer (20,480)	3	16.417 / 17.183 / 25.750	26.083 / 25.192 / 30.550	31.792 / 33.717 / 48.800	37.142 / 37.458 / 52.475	36.592 / 37.300 / 52.942
continuous MLP (20,480)	3	10.383 / 11.517 / 11.383	16.008 / 21.225 / 29.167	21.000 / 24.642 / 36.183	21.817 / 32.583 / 37.550	19.817 / 29.517 / 39.875
continuous MLP (204,800)	3	21.333 / 21.950 / 24.867	24.125 / 41.300 / 39.808	29.883 / 45.158 / 44.883	40.192 / 49.858 / 54.258	48.633 / 49.400 / 49.525
n-CLU $-$ strongest, paired	3	+4.633 / +6.917 / +3.850	+12.892 / +15.950 / +4.708	+14.725 / +15.800 / +7.125	+12.000 / +13.542 / +6.517	+11.725 / +11.558 / +7.617
3-seed mean	3	+5.133	+11.183	+12.550	+10.686	+10.300
n-CLU-context set: the rung's first 32,768 rows, exactly the n-CLU's context rows; 5,120 steps, ten times 51,200						
MLP (5,120)	3	22.07 / 25.13 / 25.90	35.79 / 41.36 / 47.81	39.32 / 43.39 / 50.93	43.64 / 47.91 / 53.64	45.77 / 48.77 / 54.26
MLP (51,200)	3	25.15 / 24.82 / 29.42	47.14 / 47.42 / 55.84	49.10 / 49.09 / 58.07	54.52 / 56.22 / 60.59	54.61 / 53.76 / 60.36
LSTM (5,120)	3	22.42 / 20.67 / 26.55	30.52 / 35.98 / 43.42	41.54 / 39.98 / 50.76	46.10 / 44.67 / 52.17	45.64 / 43.67 / 55.12
LSTM (51,200)	3	22.58 / 22.42 / 27.05	36.39 / 37.67 / 45.37	42.68 / 41.33 / 53.71	45.95 / 45.62 / 55.12	48.17 / 46.34 / 55.36
transformer (5,120)	3	13.87 / 17.50 / 25.17	27.98 / 24.17 / 29.76	32.04 / 31.27 / 47.18	35.08 / 39.42 / 53.47	36.87 / 37.83 / 52.74
transformer (51,200)	3	16.85 / 16.58 / 23.87	26.02 / 25.63 / 35.48	30.60 / 32.56 / 47.44	36.40 / 36.44 / 52.02	35.44 / 36.19 / 52.42
continuous MLP (5,120)	3	3.95 / 6.23 / 10.03	6.88 / 11.33 / 14.18	6.10 / 8.65 / 18.28	9.78 / 17.45 / 23.59	9.15 / 13.01 / 23.92
continuous MLP (51,200)	3	16.88 / 17.38 / 22.32	24.69 / 31.81 / 35.46	20.59 / 27.65 / 46.14	38.71 / 32.98 / 45.76	33.28 / 34.03 / 45.79
n-CLU $-$ strongest, paired	3	+6.65 / +7.37 / +4.22	+13.01 / +15.37 / +6.07	+16.14 / +16.73 / +9.13	+11.12 / +10.60 / +6.69	+11.30 / +12.49 / +7.29
3-seed mean	3	+6.08	+11.48	+14.00	+9.47	+10.36

4.3 Deletion on Shapes3D

After deletion the n-CLU matches an n-CLU that was never fitted on the deleted rows, and its address book matches a book refit without them: over every scope and seed the context's largest mode deviation and the relative deviation of the book's rendered features are at floating-point roundoffs, and the book's statistic does not deviate.

Conjunction deletion and the shared value: This is a measured limitation of the n-CLU. Deleting the red sphere lowers composition on the surviving combinations by about six tuples per held-out combination, on all three seeds (Fig. 9). At seed 0 the loss is concentrated where the deleted rows tied red to a shape: the combinations that request red with another shape fall by about thirty each, those that request a sphere in another color by about one, and the others don't drop. The same pipeline deleting a random 1% of items only drops on remaining requests by at most about a tenth of a tuple, so the loss is the deleted context. The drop is what a book fitted without the red sphere's rows no longer knows about red. The targeting of a deletion is a function of rank/width, and it disappears where the memory footprint is smallest. This is why refusal needs its own mechanism (Sec. B.2). Further results and discussions on deletion for the baselines and comparing that with the n-CLU are presented in App. B).

Encoder adaptation and generative baselines The frozen encoder is the primary object. Light adaptation at $K^\star$ with equal steps and images grows the n-CLU's margin over the best learned baseline on all three seeds under an anchored full unfreeze and on two of three under a low-rank adapter (Fig. 10).

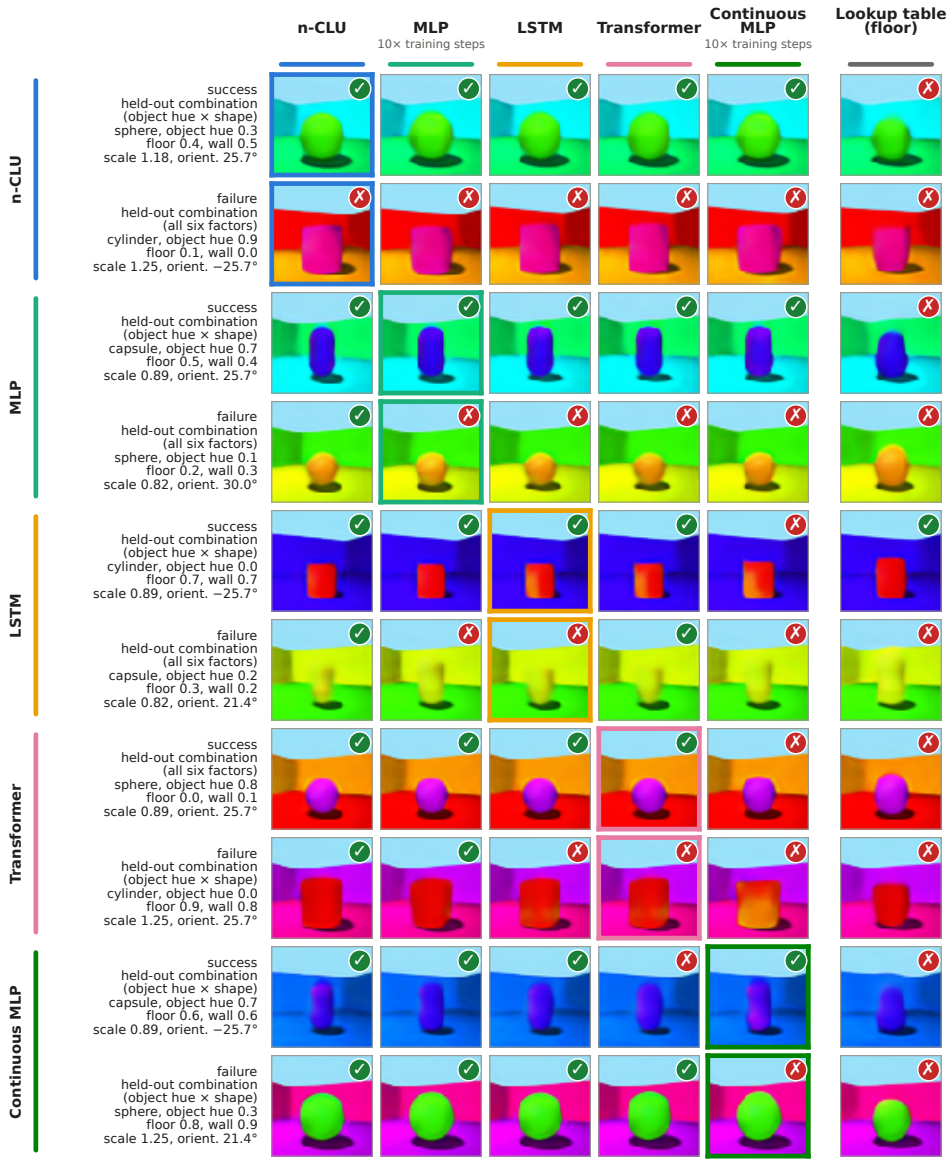


Figure 3: Each model shown succeeding and failing on held-out combinations of Shapes3D, seed 0. The model names on the left span 2 rows each, one that model got right and one wrong. Of the n-CLU's 511 invalid renders, 497 are wrong in one factor only: scale 329, orientation 125, shape 43.

4.4 CMAZE-T RESULTS AND RELATIONAL CODING

To isolate the compositional generation ability of the models, the benchmark is split into single step action prediction for a single agent given a complex rule-based condition on the answer.

CMAZE-T with symmetric inputs: On the full word design given to every arm it roughly matches the best baseline (transformer) on all three TEST seeds, as seen in Fig. 4(a). On words plus observation, on six seeds, against the transformer it has three wins, two ties and one loss. The 10-atom relational design (n-CLU-REL, below) scores every row of both tuples valid. All controls stays below the n-CLU on every seed.

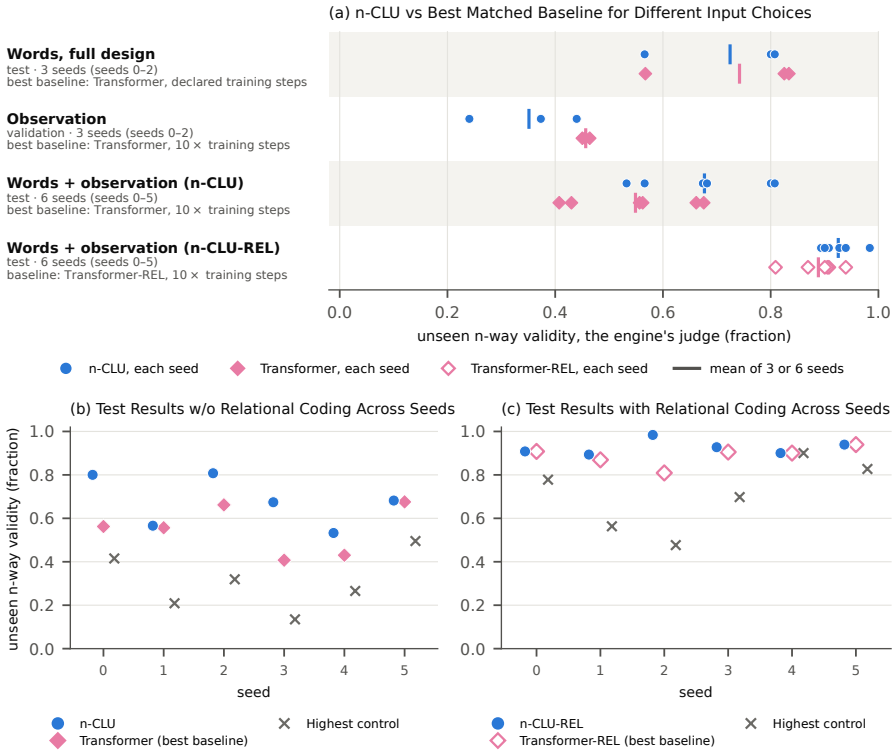


Figure 4: CMAZE-T: the n-CLU against the best matched baseline under four input choices, and six seeds without and with relational coding.(a) Unseen n-way validity under the engine's judge, one row per input choice: the full word design, the observation, and words plus observation for the n-CLU and for the n-CLU-REL. Every seed of a row is drawn: circles are the n-CLU (n-CLU-REL in the last row), diamonds the row's baseline, and a vertical tick marks the mean over the row's seeds. The baseline of the first three rows is the matched Transformer with the highest mean over the row's seeds, taken over every family and both step limits. The baseline of the n-CLU-REL row is Transformer-REL, the Transformer given the same relational columns (hollow diamonds). (b) Words plus observation, test, seeds 0–5, without relational coding: per seed, the n-CLU, the Transformer at $10\times$ training steps of the third row, and the highest of the four controls on that seed. (c) The same as (b) but with relational coding: the n-CLU-REL, the Transformer-REL of the last row, and the highest control.

Relational request coding: The relational coding is an additional set of inputs that show a map between related words, such as "tinted floor" and "blue key" are both related since they contain color as a main factor while "move East" has no color association. Fig.4(c) shows that adding the relational coding improves valid actions more than not having them but, they equally lift the transformer arm as well. Regardless, the n-CLU and n-CLU-REL are ahead of the best baseline marginally across 6 seeds.

The CMAZE-T panel figures: Figs. 5, 11–13 (App. E) draw single held-out transitions with every model's next state. Each panel is a 5×5 crop of the maze centered on the agent's cell at s_t. Walls are dark, tinted cells pale in their color, doors framed in their color with a bar across while closed, untaken keys are diamonds, the goal a cross, and the agent a black disc whose center shows the color of the key in its pocket (white when the pocket is empty). The first column is s_t with the action (an arrow, or a ring for pick-up), the second the engine's s_{t+1}, and each further column the s_{t+1} that a model's emitted outcome produces. The engine's judge does not score the picture: it scores four emitted facts, legality, the move, the key event and the pocket after the move. A picture need not show all four. So a strip under every s_{t+1} prints the emitted facts (under the engine's column, the true ones). Its first line reads legal or illegal. Its second line shows the move ($\uparrow \rightarrow \downarrow \leftarrow$, or a dot for staying), the key event ($+$ and a color for a key picked up, $-$ and a color for a key used at a door,

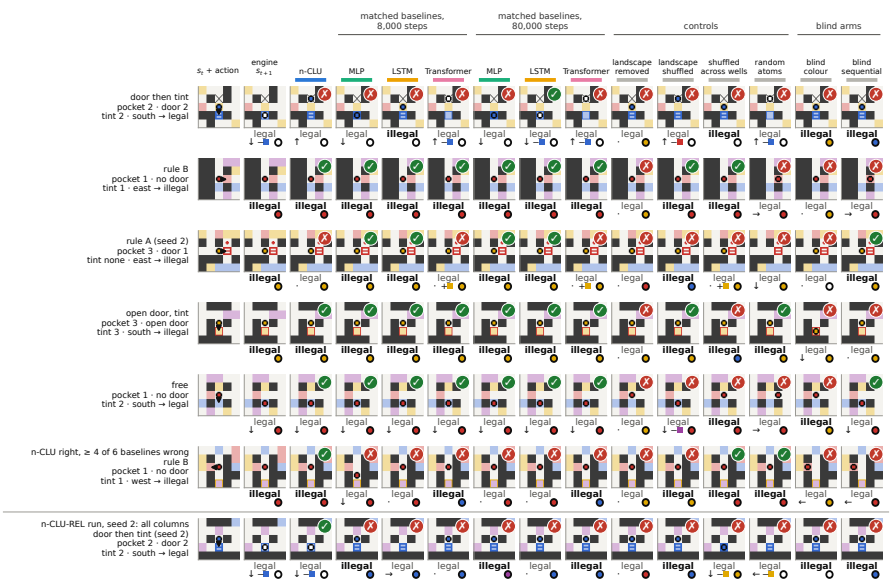


Figure 5: Held-out CMAZE-T transitions, one per rule family, with every model's next state (words plus observation, test).

blank for none) and the pocket after the move (the disc; white is empty). An illegal emission leaves the agent in place and is scored on its pocket alone, so only the pocket is shown. The tick or cross on a panel is the judge's verdict on all four facts, so two panels that show the same picture and the same strip never carry opposite marks.

5 Conclusions and Discussions

In this work we present n-CLU and demonstrate its compositional generation and deletion, matching an n-CLU never fitted on the deleted rows, at a stated partition.

The n-CLU we evaluate is the zero-dynamics limit of the CHLU (Jawahar & Pierini, 2026): the same context, wells and read, with the settling loop's step count set to zero. Inside a well this is an algebraic identity: the n-CLU's energy is the distance to the atom span, the request's basin sits at an address inside that span, so both minima coincide and an in-well settle has nowhere to go (derivation in App. D).

We show performance on Shapes3D that scores above all matched baselines by ∼seven new tuples per held-out combination over multiple seeds.

We also create and provide a small scale dataset CMAZE-T targeted at world models and their valid, composition abilities. CMAZE-T is the first instrument of a planned CMAZE-suite on the same engine and judge. The transition task will be extended to observation generation, to route planning and to the further tasks already prototyped, so that CMAZE as a whole becomes a useful benchmark for the world-models community. On CMAZE-T n-CLU's lead over matched baselines is small and seed-dependent.

The main limitations we plan to expand upon are improving the memory footprint required for deletion, expanding the CMAZE-suite to more complex, long-horizon cases, and evaluating the n-CLU on real world-model relevant data.

The follow-up is a full evaluation of the settling (physics) CLU on the three properties of Sec. 1.1, same judges, memory footprint and matching, with this paper's n-CLU as its zero-dynamics ablation. Our future work also includes improving the memory footprint when moving to the physics version. We also intend to use the n-CLU as a diagnostic tool to understand how best to construct the latent physics-CLU.

AI USE STATEMENT

In this work agentic coding tools were used to draft and automate the code based experiments. The code was verified using human-authored code harnesses (as a unit-test style quality control). The results generated were evaluated and validated by human authors. Generative AI was used to draft the initial skeleton of this manuscript with human authors responsible for line-by-line major revisions and edits of the text and plots and other authors provided editorial passes on the human-revised version, blind to the initial AI generated draft. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Here is how we used AI tools for this work:

Generate synthetic dataset (CMAZE-T; agentic AI to write the simulation code and then ran it manually), help develop theoretical models or conceptual frameworks (n-CLU to physics-CLU limits), formulate mathematical claims (physics-CLU related proof), provide critical ingredients for proving mathematical claims (physics-CLU), assist in the writing of proofs (physics-CLU), implement methods (coding, code setup, scaffolding), clean and reformat dataset (for Shapes3D via opensource code, CMAZE-T via custom code), support qualitative and thematic data analysis (plot generation via code), create or modify scientific figures or images (plot generation via code), create or edit software code, draft parts of a research paper, summarize or analyze existing literature, sourcing/searching for information, identify relevant literature, propose keywords for a research paper.

ETHICS STATEMENT

The study uses a public synthetic image dataset (Shapes3D) and procedurally generated mazes; it involves no human subjects and no personal data. Deletion is relevant to data-protection requests; we claim only that the n-CLU and its address book match those never fitted on the deleted rows (the frozen encoder and decoder are outside the object), no certified unlearning is to be inferred, and we state in Sec. 4.3 a case in which deleting a conjunction changes what the model answers about the values that survive it.

REPRODUCIBILITY STATEMENT

Each number in this paper is read from the recorded result of a declared run, and nearly every one carries its provenance beside it in the source. The verbatim re-check against the cited records is partial: an automated check covers every number of three groups of runs and, overall, 721 of the 733 sourced number tokens of this draft, and it checks that a number is present in its record, not what the number means. The missing numbers are likely the rounded off estimates and the broad numbers from App.D, that were from another code base which is not provided here since these are not main text numbers and are only meant as a preliminary motivation of our future plans for the n-CLU (this paper) to physics-CLU (future work) transition. The object, judge, estimand and memory footprint are defined in Sec. 2; every table states its seeds, condition and pins. The anonymised code archive:`https://anonymous.4open.science/r/n-clu-F746/`.

REFERENCES

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine Unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159, San Francisco, CA, USA, May 2021. IEEE. ISBN 978-1-7281-8934-5. doi: 10.1109/SP40001.2021.00019. URL `https://ieeexplore.ieee.org/document/9519428/`.

Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

Yinzhi Cao and Junfeng Yang. Towards Making Systems Forget with Machine Unlearning. In *2015 IEEE Symposium on Security and Privacy*, pp. 463–480, San Jose, CA, May 2015. IEEE. ISBN 978-1-4673-6949-7. doi: 10.1109/SP.2015.35. URL `https://ieeexplore.ieee.org/document/7163042/`.

Yilun Du, Shuang Li, and Igor Mordatch. Compositional Visual Generation and Inference with Energy Based Models, December 2020. URL http://arxiv.org/abs/2004.06030. arXiv:2004.06030 [cs.CV].

Markos Georgopoulos, Grigorios Chrysos, Maja Pantic, and Yannis Panagakis. Multilinear Latent Conditioning for Generating Unseen Attribute Combinations, September 2020. URL http://arxiv.org/abs/2009.04075. arXiv:2009.04075 [cs.LG].

Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making AI Forget You: Data Deletion in Machine Learning, November 2019. URL http://arxiv.org/abs/1907.05012. arXiv:1907.05012 [cs.LG].

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9301–9309, Seattle, WA, USA, June 2020. IEEE. ISBN 978-1-7281-7168-5. doi: 10.1109/CVPR42600.2020.00932. URL https://ieeexplore.ieee.org/document/9157084/.

Aditya Golatkar, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Training Data Protection with Compositional Diffusion Models, October 2024. URL http://arxiv.org/abs/2308.01937. arXiv:2308.01937 [cs.LG].

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified Data Removal from Machine Learning Models, November 2023. URL http://arxiv.org/abs/1911.03030. arXiv:1911.03030 [cs.LG].

David Ha and Jürgen Schmidhuber. World Models. March 2018. doi: 10.5281/ZENODO.1207631. URL https://zenodo.org/record/1207631.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, April 2025. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-025-08744-2. URL https://www.nature.com/articles/s41586-025-08744-2.

Geoffrey E. Hinton. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8):1771–1800, August 2002. ISSN 0899-7667, 1530-888X. doi: 10.1162/089976602760128018. URL https://direct.mit.edu/neco/article/14/8/1771-1800/6687.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997. ISSN 0899-7667, 1530-888X. doi: 10.1162/neco.1997.9.8.1735. URL https://direct.mit.edu/neco/article/9/8/1735-1780/6109.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering Interpretable GAN Controls, December 2020. URL http://arxiv.org/abs/2004.02546. arXiv:2004.02546 [cs.CV].

Pratik Jawahar and Maurizio Pierini. Chlu: The causal hamiltonian learning unit as a symplectic primitive for deep learning. *arXiv preprint arXiv:2603.01768*, 2026.

Hyunjik Kim and Andriy Mnih. Disentangling by Factorising, July 2019. URL http://arxiv.org/abs/1802.05983. arXiv:1802.05983 [stat.ML].

Zhifeng Kong and Scott Alfeld. Approximate Data Deletion in Generative Models, June 2022. URL http://arxiv.org/abs/2206.14439. arXiv:2206.14439 [cs.LG].

Zhifeng Kong and Kamalika Chaudhuri. Data Redaction from Pre-trained GANs. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 638–677, Raleigh, NC, USA, February 2023. IEEE. ISBN 978-1-6654-6299-0. doi: 10.1109/SaTML54575.2023.00048. URL https://ieeexplore.ieee.org/document/10136171/.

Sébastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien. Additive Decoders for Latent Variables Identification and Cartesian-Product Extrapolation. In *Advances in Neural Information Processing Systems 36*, pp. 25112–25150, New Orleans, Louisiana, USA, 2023. Neural Information Processing Systems Foundation, Inc. (NeurIPS). ISBN 978-1-7138-9911-2. doi: 10.52202/075280-1092. URL http://www.proceedings.com/075280-1092.html.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional Visual Generation with Composable Diffusion Models. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, volume 13677, pp. 423–439. Springer Nature Switzerland, Cham, 2022. ISBN 978-3-031-19789-5 978-3-031-19790-1. doi: 10.1007/978-3-031-19790-1_26. URL https://link.springer.com/10.1007/978-3-031-19790-1_26. Series Title: Lecture Notes in Computer Science.

Milton Montero, Jeffrey Bowers, Rui Ponte Costa, Casimir Ludwig, and Gaurav Malhotra. Lost in Latent Space: Examining Failures of Disentangled Models at Combinatorial Generalisation. In *Advances in Neural Information Processing Systems 35*, pp. 10136–10149, New Orleans, Louisiana, USA, 2022. Neural Information Processing Systems Foundation, Inc. (NeurIPS). ISBN 978-1-7138-7108-8. doi: 10.52202/068431-0736. URL http://www.proceedings.com/068431-0736.html.

Saemi Moon, Seunghyuk Cho, and Dongwoo Kim. Feature Unlearning for Pre-trained GANs and VAEs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21420–21428, March 2024. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v38i19.30138. URL https://ojs.aaai.org/index.php/AAAI/article/view/30138.

Weili Nie, Arash Vahdat, and Anima Anandkumar. Controllable and Compositional Generation with Latent-Space Energy-Based Models, December 2021. URL http://arxiv.org/abs/2110.10873. arXiv:2110.10873 [cs.CV].

James Oldfield, Markos Georgopoulos, Yannis Panagakis, Mihalis A Nicolaou, and Ioannis Patras. Tensor Component Analysis for Interpreting the Latent Space of GANs. In *Proceedings of the British Machine Vision Conference 2021*, pp. 430, Online, 2021. British Machine Vision Association. ISBN 978-1-901725-68-1. doi: 10.5244/C.35.430. URL https://www.bmva.org/bmvc/2021/conference/papers/paper_1617.html.

James Oldfield, Christos Tzelepis, Yannis Panagakis, Mihalis A. Nicolaou, and Ioannis Patras. PandA: Unsupervised Learning of Parts and Appearances in the Feature Maps of GANs, February 2023. URL http://arxiv.org/abs/2206.00048. arXiv:2206.00048 [cs.CV].

Yijun Quan, Wentai Wu, and Giovanni Montana. Exact Federated Continual Unlearning for Ridge Heads on Frozen Foundation Models. In Michelangelo Ceci, Dragi Kocev, Anna Monreale, Celine Vens, Peter Flach, Giuseppe Manco, Myra Spiliopoulou, and Elio Masciari (eds.), *Machine Learning and Knowledge Discovery in Databases. Research Track*, volume 16946, pp. 114–131. Springer Nature Switzerland, Cham, 2027. ISBN 978-3-032-37672-5 978-3-032-37673-2. doi: 10.1007/978-3-032-37673-2_7. URL https://link.springer.com/10.1007/978-3-032-37673-2_7. Series Title: Lecture Notes in Computer Science.

Hui Sun, Tianqing Zhu, Wenhan Chang, and Wanlei Zhou. Generative Adversarial Networks Unlearning. *IEEE Transactions on Dependable and Secure Computing*, 22(5):5303–5320, September 2025. ISSN 1545-5971, 1941-0018, 2160-9209. doi: 10.1109/TDSC.2025.3564992. URL https://ieeexplore.ieee.org/document/10979463/.

Joshua B. Tenenbaum and William T. Freeman. Separating Style and Content with Bilinear Models. *Neural Computation*, 12(6):1247–1283, June 2000. ISSN 0899-7667, 1530-888X. doi: 10.1162/089976600300015349. URL https://direct.mit.edu/neco/article/12/6/1247-1283/6381.

Keyon Vafa, Justin Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. Evaluating the World Model Implicit in a Generative Model. In *Advances in Neural Information Processing Systems 37*, pp. 26941–26975, Vancouver, BC, Canada, 2024. Neural Information Processing Systems Foundation, Inc. (NeurIPS). ISBN 979-8-3313-1438-5. doi: 10.52202/079017-0846. URL http://www.proceedings.com/079017-0846.html.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion Models Are Real-Time Game Engines, April 2025. URL http://arxiv.org/abs/2408.14837. arXiv:2408.14837 [cs.LG].

M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear Analysis of Image Ensembles: TensorFaces. In Gerhard Goos, Juris Hartmanis, Jan Van Leeuwen, Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen (eds.), *Computer Vision — ECCV 2002*, volume 2350, pp. 447–460. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002. ISBN 978-3-540-43745-1 978-3-540-47969-7. doi: 10.1007/3-540-47969-4_30. URL http://link.springer.com/10.1007/3-540-47969-4_30. Series Title: Lecture Notes in Computer Science.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, August 2023. URL http://arxiv.org/abs/1706.03762. arXiv:1706.03762 [cs.CL].

Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. *ACM Transactions on Graphics*, 24(3):426–433, July 2005. ISSN 0730-0301, 1557-7368. doi: 10.1145/1073204.1073209. URL https://dl.acm.org/doi/10.1145/1073204.1073209.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional Generalization from First Principles. In *Advances in Neural Information Processing Systems 36*, pp. 6941–6960, New Orleans, Louisiana, USA, 2023. Neural Information Processing Systems Foundation, Inc. (NeurIPS). ISBN 978-1-7138-9911-2. doi: 10.52202/075280-0304. URL http://www.proceedings.com/075280-0304.html.

Mimee Xu, Jiankai Sun, Xin Yang, Kevin Yao, and Chong Wang. Netflix and Forget: Efficient and Exact Machine Unlearning from Bi-linear Recommendations, February 2023. URL http://arxiv.org/abs/2302.06676. arXiv:2302.06676 [cs.LG].

Linfeng Zhao, Lingzhi Kong, Robin Walters, and Lawson L. S. Wong. Toward Compositional Generalization in Object-Oriented World Modeling, June 2022. URL http://arxiv.org/abs/2204.13661. arXiv:2204.13661 [cs.LG].

APPENDICES

A THE K-CURVE: COMPOSITION AND THE MEMORY FOOTPRINT AGAINST K

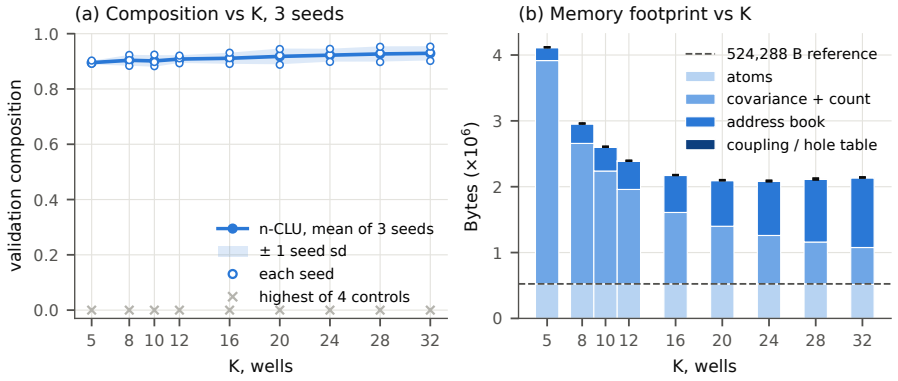


Figure 6: The K-sweep validation, 3 seeds at every K. (a) Validation composition against K: the n-CLU's mean over seeds 0, 1 and 2 with a band of one seed standard deviation, each seed marked, and the highest of the four controls (landscape removed, landscape shuffled, shuffled across wells, random atoms). (b) The memory footprint in its four parts (atoms, covariance and count, address book, coupling and hole table; symmetric-triangle convention). The values are printed in Tab. 4.

Table 3: The generation table at $K^\star = 32$. Outperforms LACE (Nie et al., 2021), PoE (Hinton, 2002)). Caveats mentioned below. The nearest-prototype row is a ceiling, not a baseline: a per-factor prototype table built from the factor labels of every training row and decoded directly through an unbounded real-valued channel, selected under an earlier byte limit (declared). It beats $K^\star$ on the per-factor aggregate shown here, but on exact whole-request validity, the metric of Tab. 1, $K^\star$ beats it on every seed (0.50 / 0.99 / 0.50 against 0.17 / 0.14 / 0.29; the prototype table gets scale wrong). The deployed read is the object of study because it is label-free in its content and deletable by downdate; the prototype table is neither.

arm	seeds	aggregate, seed 0 / 1 / 2	mean	deployed bytes (convention)
$K^\star = 32$	3	**0.5718 / 0.9977 / 0.6806**	**0.7500 ± 0.1807**	2,145,738 / 2,153,018 / 2,155,618 B, four-part, never netted (4.09–4.11× ref)
landscape removed	3	0.0556 / 0.0556 / 0.2778	0.1296	same object, atoms zeroed
landscape shuffled	3	0.1481 / 0.1389 / 0.0602	0.1157	same bytes
shuffled across wells	3	0.1065 / 0.0764 / 0.1435	0.1088	same bytes
random atoms	3	0.1458 / 0.2523 / 0.1597	0.1860	same bytes
earlier n-CLU's read	3	0.5903 / 0.6065 / 0.5880	0.5949 ± 0.0082	524,288 B (n-CLU modes)
n-CLU removed	3	0.1111 / 0.1389 / 0.1111	0.1204	524,288 B
nearest-prototype ceiling	3	0.8588 / 0.8542 / 0.8819	0.8650 ± 0.0122	475,136 B (selected under a retired byte limit, declared)
LACE (`ode`)	3	0.2940 / 0.3241 / 0.2917	0.3032 ± 0.0148	not byte-matched
PoE (`ebm_compose`)	3	0.1759 / 0.1875 / 0.2269	0.1968 ± 0.0218	14,226,660 B (27.1× ref)

LACE and PoE: LACE (Nie et al., 2021) and PoE (Hinton, 2002) were scored beside the learned baselines on rows 3–4 and on every ladder rung (Tab. 3), 3 seeds. No learned-baseline column was replaced. They do not beat the n-CLU on rows 3–4, nor anywhere on the ladder: on rows 3–4 the $K^\star$ n-CLU scores 47.50 / 47.40 / 49.12 novel valid tuples per held-out combination against LACE 0.165 / 0.073 / 0.229 and PoE 0.009 / 0.000 / 0.000, and the n-CLU wins 109/109 combinations against each, on every seed; on every rung it wins every combination against both on every seed (60 of 60 at $n = 2$, 120 of 120 at $n \geq 3$). Their settings are their own and are not byte-matched: LACE 4,073,663 classifier parameters (31.08× the 524,288 B reference; 63.10× counting its whitening and mean), PoE six experts with 3,556,665 parameters (27.135×), each fitted on 32,256 rows with 512 validation rows from the rung's own pool and 36 selection trials. *This is LACE and PoE as instantiated in this feature space at their own declared settings, it is not a refutation of LACE given there is no generative baseline.*

Table 4: The K-curve at 3 seeds on every row, on the validation set.

K	seeds	rank/width	val comp. mean	seed sd	drop vs K=5	four-part B mean	seed sd B	strict check
5	3	0.1561–0.1565	0.8953	0.0065	+0.0000	4,118,034	0	pass
8	3	0.2500–0.2500	0.9037	0.0197	−0.0085	2,959,602	0	pass
10	3	0.3122–0.3137	0.9014	0.0211	−0.0062	2,606,754	0	pass
12	3	0.3743–0.3765	0.9082	0.0141	−0.0129	2,393,714	0	pass
16	3	0.5000–0.5000	0.9111	0.0203	−0.0158	2,177,330	0	pass
20	3	0.5294–0.6275	0.9178	0.0285	−0.0226	2,098,879	2,864	**fail**
24	3	0.4353–0.7529	0.9223	0.0235	−0.0270	2,086,677	7,060	pass
28	3	0.3425–0.8767	0.9267	0.0276	−0.0314	2,117,183	9,885	pass
32	3	0.2656–0.9844	0.9290	0.0259	−0.0338	2,140,538	3,640	pass

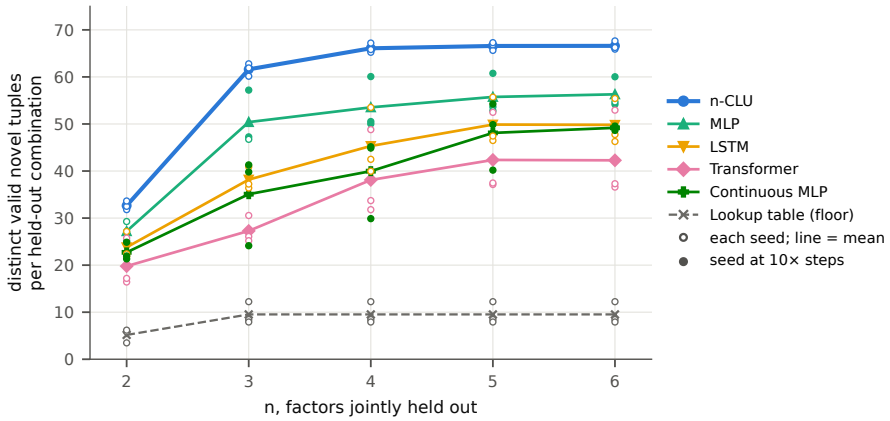


Figure 7: The ladder at $K^\star$ against matched baselines on Shapes3D, across 3 seeds showing the distinct valid new tuples created per held-out combination against n, the number of factors jointly held out.

B Deletion: Extended Discussion

B.1 Deletion against retrained baselines

A learned baseline's only route to a model never fitted on the rows is to retrain from scratch without them. We ran that route for every data-matched baseline family on the same deleted rows of the same pool, at the baseline's selected schedule, three seeds (Tabs. 5 and 6, which print each retrain's rows read and time. After the value and item deletions the n-CLU stays ahead of every retrained baseline on every seed, with every 95% interval above zero. After the red-sphere deletion it is ahead on the mean at every seed, but its interval crosses zero against the MLP at seeds 1 and 2 and against the LSTM at seed 2 (Fig. 9): its own loss on that deletion, about six tuples per held-out combination, is larger than the change any retrained baseline shows except the continuous MLP at seed 2, so on collateral damage the baselines' crude retrain is cleaner than the n-CLU's deletion. Nor is the n-CLU's route faster. Its downdate takes seconds, but the book's refit takes longer than any baseline's retrain. What the n-CLU's route has that the retrain does not is the match with a refit without the rows, by construction, and no pool row read. The baselines retrain at their selected schedule, so the MLP (seeds 0 and 1) and continuous-MLP convergence flags of Tab. 1 apply to them.

Damaged renders converging to sphere: A damaged render could be judged sphere for rendering reasons and score spuriously on the deleted conjunction. On the trained n-CLU's presented here, a degraded render is judged cube, the held-out shape, so the judged-sphere excess is negative for both maximally degraded arms (-0.3291 for landscape removed, -0.3078 for random atoms) against $+0.0074$ after the red-sphere deletion.

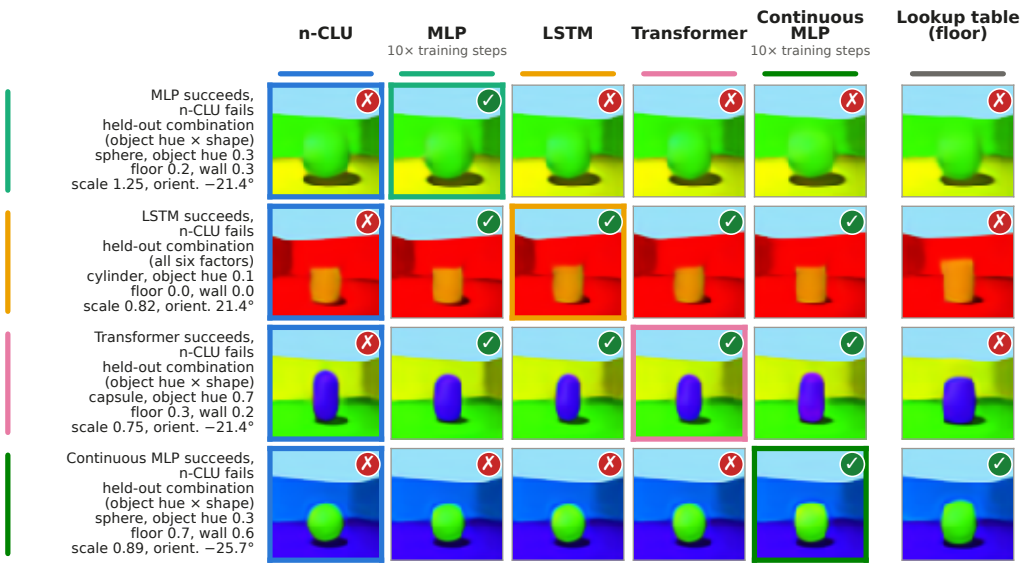


Figure 8: Held-out requests on which a baseline succeeds and the n-CLU fails (seed 0).

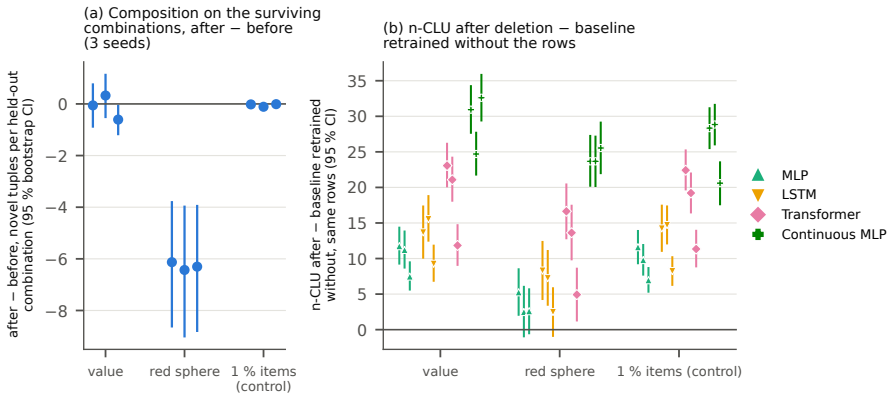


Figure 9: Deletion at $K^\star = 32$, rows 3–4, test set, fit pool 131,072 rows, seeds 0, 1 and 2. Deleting a value (red), a conjunction (red sphere) or a random 1% of items downdates the n-CLU's context, downdates the address book's statistic and refits the book from it, and punches the hole. (a) Composition on the remaining held-out combinations (89 for the value, 109 for the others), after minus before, per seed, with its 95% bootstrap interval. (b) The n-CLU after deletion minus each baseline retrained without the same rows of the same pool, with 95% intervals: above zero for every baseline and seed on the value and item deletions; on red sphere the interval includes zero against the MLP at seeds 1 and 2 and against the LSTM at seed 2.

B.2 Refusal through the hole:

Refusal is a second mechanism on the same object: a coupling table with *holes*. Punching a hole for a deleted value or conjunction marks the request slot as refused. On the $K^\star$ object two holes cost a byte each, read no data rows, take no measurable time and refit nothing, in a coupling table whose layout is asserted equal to the field-mode n-CLU's over every pair slot. One hole (red $\wedge$ cube) is refusable although no red cube was ever written: refusal does not need a deleted row to have existed. The hole is the only mechanism on this object that refuses, and it is a flag: a request for a holed value is refused by construction, so refusal is measurement (c) Symbolic refusal is not claimed. An item deletion makes nothing impossible: on the n-CLU's own pool, deleting a random 1% of items clears

Table 5: Deletion cost for the data-matched baselines: retrain without the rows, on the n-CLU's 131,072-row pool, at the selected schedule, seeds 0 / 1 / 2.

baseline, retrain without	seeds	value (red), s	red sphere, s	item (1%), s	rows read	bytes rewritten
MLP	3	547.7 / 677.4 / 729.8	814.3 / 805.6 / 746.6	884.6 / 922.8 / 742.5	120,028 / 125,561 / 129,761	522,056
LSTM	3	757.5 / 712.4 / 1,016.0	772.8 / 767.5 / 1,198.2	773.2 / 1,093.8 / 1,034.6	120,028 / 125,561 / 129,761	522,884
transformer	3	1,298.0 / 1,772.2 / 1,705.3	1,255.8 / 1,987.7 / 2,064.5	1,246.0 / 1,754.9 / 1,562.8	120,028 / 125,561 / 129,761	521,732
continuous MLP	3	695.9 / 760.1 / 673.3	922.3 / 859.6 / 712.3	817.2 / 862.4 / 629.7	120,028 / 125,561 / 129,761	527,564

Table 6: The n-CLU after deletion against each baseline retrained without the same rows, all 109 combinations, seeds 0 / 1 / 2. Baseline: as trained → retrained without. n-CLU − baseline: the paired difference after deletion, with the 95% bootstrap interval where it crosses zero or comes closest to it. The n-CLU after deletion reads 50.865 / 50.854 / 52.000 (value, survivor combinations), 41.367 / 40.972 / 42.817 (red sphere) and 47.477 / 47.294 / 49.110 (item).

scope	baseline	seeds	baseline, as trained → retrained without (0 / 1 / 2)	n-CLU − baseline after deletion (0 / 1 / 2)
value	MLP	3	39.607 → 39.124 / 40.449 → 39.640 / 46.315 → 44.528	+11.742 / +11.213 / +7.472
	LSTM	3	36.685 → 37.191 / 34.831 → 35.315 / 43.303 → 42.742	+13.674 / +15.539 / +9.258
	transformer	3	27.933 → 27.798 / 30.090 → 29.775 / 40.865 → 40.157	+23.067 / +21.079 / +11.843
	continuous MLP	3	21.393 → 19.933 / 20.213 → 26.157 / 29.652 → 19.393	+30.933 / +24.697 / +32.607
red sphere	MLP	3	36.706 → 36.110 / 37.569 → 38.486 / 43.174 → 40.220	+5.257 / **+2.486 [**−0.94**, 6.02]** / **+2.596 [**−0.50**, 5.67]**
	LSTM	3	33.972 → 33.064 / 32.358 → 33.734 / 40.697 → 40.339	+8.303 / +7.239 / **+2.477 [**−0.87**, 5.82]**
	transformer	3	25.560 → 24.725 / 27.789 → 27.349 / 38.651 → 37.890	+16.642 / +13.624 / +4.927 [1.29, 8.57]
	continuous MLP	3	17.917 → 17.697 / 16.541 → 17.303 / 26.844 → 17.275	+23.670 / +23.670 / +25.541
item	MLP	3	36.706 → 35.881 / 37.569 → 37.505 / 43.174 → 42.156	+11.596 / +9.789 / +6.954 [5.34, 8.66]
	LSTM	3	33.972 → 33.248 / 32.358 → 32.587 / 40.697 → 40.881	+14.229 / +14.706 / +8.229
	transformer	3	25.560 → 25.055 / 27.789 → 28.083 / 38.651 → 37.771	+22.422 / +19.211 / +11.339
	continuous MLP	3	17.917 → 19.138 / 16.541 → 18.450 / 26.844 → 28.514	+28.339 / +28.844 / +20.596

no value bit and punches no hole, so no request becomes certifiably impossible and the item row has no refusal row at all.

C Light Encoder Adaptation

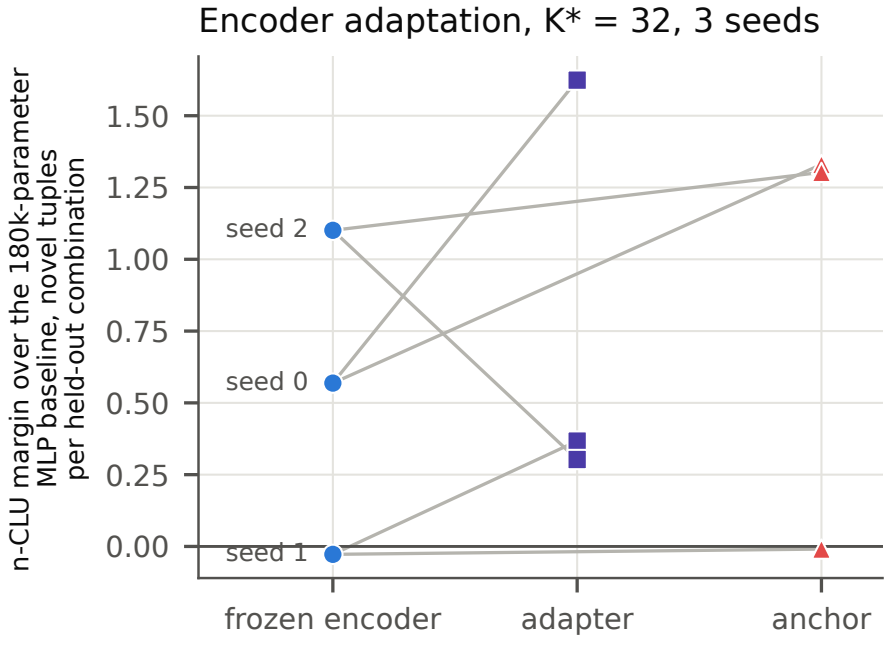


Figure 10: Encoder adaptation margins: The n-CLU minus the 180k-parameter MLP baseline, in novel tuples per held-out combination, per seed at $K^\star$, with the frozen encoder, an adapter and an anchor.

D n-CLU is a special case of a Physics motivated CLU

The derivation: The context is an orthonormal basis U_k of the per-well atom span, and the read renders the request's address through that basis, $x_k = U_k c_k$. The n-CLU's energy for a basis is the distance to its span, $E_{\text{context},k} = \frac{1}{2}\,\text{dist}^2(x_k, \text{span}\,G_k)$, zero everywhere on the span; the request enters as a basin at the address, $E_{\text{req},k} = \frac{1}{2}\beta\|x_k - a_k\|^2$, zero at the address. Because $a_k \in \text{span}(G_k)$ by construction of the renderer, both terms attain their minima at the same point, so an in-well settle has nowhere to go.

The four conditions: (C1) The n-CLU's energy is distance to the atom span, the atoms being a basis: with an atoms-as-points energy on the same context the settle moves (relative displacement median 1.7244, max 2.6682 on a synthetic check). (C2) The address lies in the span: pushed off the span by $\delta = 0.10$ the settle moves by exactly $\delta/(1+\beta)$, measured 0.050000 against the closed form 0.050000. (C3) The state is carried in span coordinates. The same energy in channel coordinates drifts by 7.447×10^{-08} relative, float32 rounding, so the paper says "identical" of the channel form and "bit-identical" only of the span form. (C4) No cross-well term: at $\lambda = 0$ the displacement is exactly 0.0; at $\lambda = 10^{-4}$, 10^{-2}, 0.05, 0.2 it is 7.875×10^{-05}, 7.882×10^{-03}, 3.971×10^{-02}, 1.720×10^{-01}. Two wells of the deployed n-CLU were full rank, where an in-well settle is trivially a no-op. The coercivity certificate reproduces from first principles: $\lambda^\star = 0.25000000$ and a certificate minimum eigenvalue of -1.50000000 at the declared $\lambda = 1$, so the landscape is unbounded below at the declared strength and the used strength is 0.225000. The identity is in-well only; a $\lambda = 0$, steps> 0 arm was never run on the deployed n-CLU (the synthetic check covers steps 0 / 1 / 10 / 1000); and the 1.9×10^{-7} off-span residual is the renderer's float32 rounding, the same size for the static n-CLU that never moves.

E Maze world-model rows

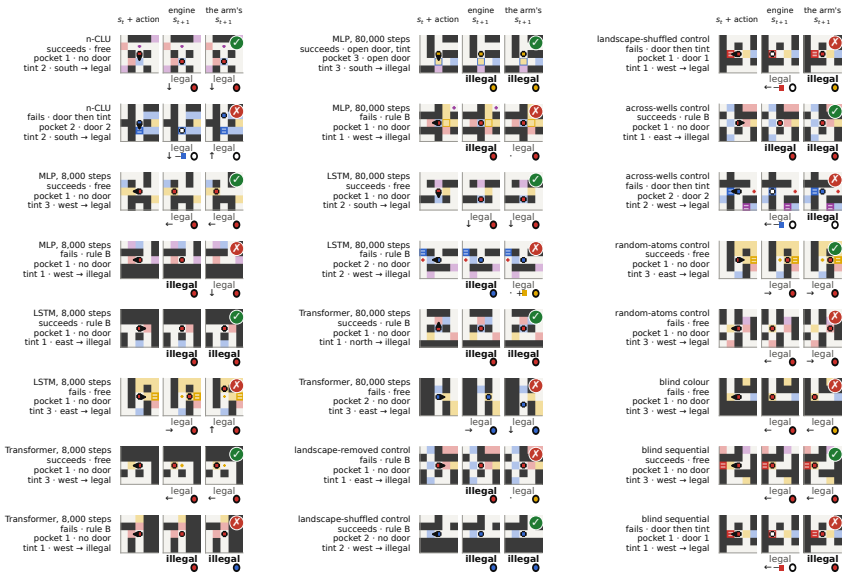


Figure 11: Each CMAZE-T model succeeding and failing (words plus observation, test, seed 0). For every model, one held-out n-way transition on which its s_{t+1} is correct and one on which it is not, each drawn uniformly with a fixed seed among the qualifying transitions. The landscape-removed control and the blind-color arm are correct on no held-out n-way transition at seed 0 and show only a failure. The n-CLU has no door success at seeds 0–2 (Fig. 5), so its success row is the uniform draw. Columns: s_t with the action, the engine's s_{t+1}, and the model's s_{t+1} with the engine's judge's mark. The strip under each s_{t+1} is read as described in Sec. 4.4.



Figure 12: Each CMAZE-T model succeeding and failing with relational coding (words plus observation, test). As Fig. 11, for the n-CLU-REL, the matched baselines, the controls and the blind arms of the same runs, and for MLP-REL, LSTM-REL and Transformer-REL at 8,000 and 80,000 training steps (the baselines given the same relational columns). All rows are from seed 0 except the n-CLU-REL success row, which is a door success drawn by rule among the 44 at seed 2 (Fig. 5). Every panel is re-read from the runs' persisted models; the re-read reproduces all 132 test scores of seeds 0–5. The strip under each s_{t+1} is read as described in Sec. 4.4.

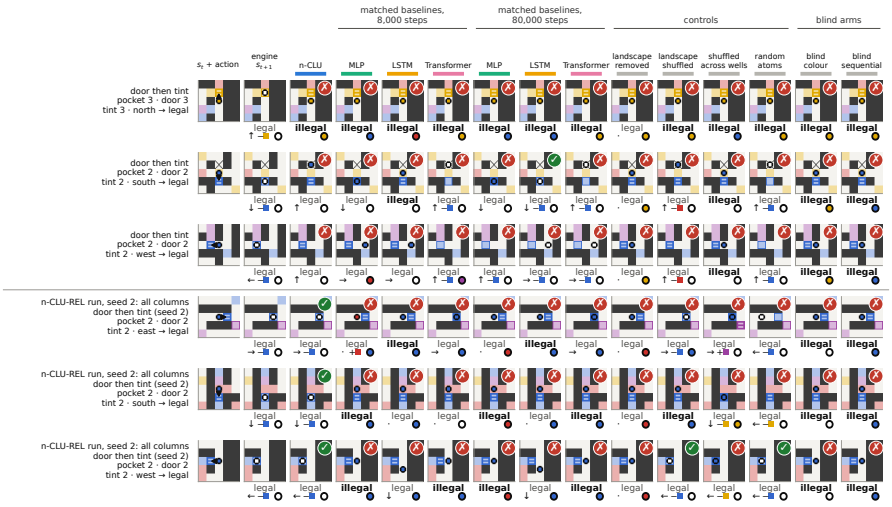


Figure 13: Door-then-tint transitions: the move opens a door with the key in the pocket and enters a tint of the same color (words plus observation, test). Columns as in Fig. 5. Rows 1–3: the first such held-out transition at seed 0 for each move direction present (north, south, west); the n-CLU is wrong on all three, as on every one of its door-then-tint transitions at seeds 0–2. Rows 4–6: door successes drawn by rule from the n-CLU-REL at seed 2, one per move direction among its successes (east, south, west; it is wrong on every northward one), with every column of these rows from that run; fixed-seed draw (seed 16). The strip under each s_{t+1} is read as described in Sec. 4.4.