Template Priors in Small CNNs: Learning Dynamics and the Measurement Dependence of Causal Usefulness

Anonymous authors
Paper under double-blind review

Abstract

Template alignment reveals what a convolutional kernel resembles, but does not establish how the network uses its features. We study this distinction between kernel appearance and causal usefulness in small CNNs on two controlled rendering tasks. A 400-run experiment increases template alignment but does not establish an improvement in selected-channel intervention fidelity under four Holm-corrected primary tests. A subsequent 200-run experiment extends training and gradually releases the retention penalty. Template initialization does not uniformly accelerate learning. Releasing retention improves compositional accuracy relative to constant retention while reducing alignment. However, the relative causal-usefulness difference depends on how it is measured. A reanalysis of 80 existing final checkpoints tests three validation-only channel rankings and four intervention sizes. On two_concepts in TinyCNN, every ranking yields negative release effects at one or two channels and positive effects at four or eight; TwoLayerCNN outcomes also depend on ranking and size. A further analysis of 2,000 saved checkpoint states shows that the retained-kernel contrast persists under one-to-one kernel matching. We connect template constraints, learning dynamics and intervention design: a preserved kernel shape and a single patching score are insufficient to establish an interpretability benefit. The later comparisons are exploratory, and the results do not establish general human interpretability or causal identifiability.

1 Motivation and research questions

An edge, corner or ring can remain recognizable in a convolutional kernel after training. What does that appearance tell us about the network's computation? Template alignment answers a question about the weights; concept selectivity and activation patching ask different questions about the features and their effects on predictions. The distinction matters when structured convolutional priors are used to make learned features easier to inspect.

Prior research provides methods to initialize, parameterize or freeze filters, as well as methods for measuring concept alignment and intervening on units. We study what happens when a template constraint is retained or gradually released during training. Our central finding is that the retention–release comparison can reverse when the intervention size changes, even though the difference in kernel appearance persists. This makes the choice of measurement part of the interpretation of the result.

We ask whether greater retained template alignment accompanies independently measured concepts and useful matched interventions, whether an early learning advantage persists with longer optimization, and whether a comparative causal conclusion survives changes in selection and intervention size. We operationalize these questions using two related rendering tasks with known concepts and matched counterfactuals. Low-level kernel alignment is kept distinct from object-level concept selectivity and localization.

The study proceeds sequentially. A 400-run experiment fixes four primary contrasts before outcome inspection. Its unresolved causal result and frozen-head diagnostic motivate a longer 200-run experiment on fresh blocks. A subsequent 80-checkpoint reanalysis tests whether the new four-channel comparison depends on its measurement. These stages have different statistical status and are not pooled as independent replications of one effect.

Our contributions are: (i) a controlled comparison of template strength with renderer-based concept and intervention measures, including matched normalization and spectrum controls in the initial stage; (ii) longitudinal evidence separating initialization, continued retention and release; and (iii) a documented reversal of the relative release effect across channel budgets under several validation-only rankings. We do not propose a new general filter family, claim that retained kernels are human-interpretable by construction, or claim priority for the general observation that patching choices matter.

2 Related work and contribution boundary

2.1 Structured and fixed convolutional filters

Gabor initialization is a direct antecedent: Özbulak and Ekenel initialize a CNN's first layer with a Gabor bank (Özbulak & Ekenel, 2018). Molaei and Shiri Ahmad Abadi explicitly address the loss of Gabor structure by learning its generating parameters, rather than freely updating every filter entry (Molaei & Shiri Ahmad Abadi, 2020). PCFNet similarly replaces first-layer kernels with learnable predefined image filters (Ma et al., 2020). Wang and Alkhalifah apply constrained learnable Gabor kernels to seismic facies classification (Wang & Alkhalifah, 2024). Consequently, neither starting with interpretable-looking kernels nor preserving a structured family during learning is our novelty. Our edge/corner/ring anchors and soft weight-space penalty also differ from an exact Gabor parameterization: an anchored filter can leave the template family, and release explicitly removes that penalty.

Fixed filters need not preclude strong prediction. Linse, Barth and Martinetz keep spatial filters fixed and learn pointwise combinations in PFCNNs (Linse et al., 2023). ExplainFix studies spatially fixed initialization, steering and pruning (Gaudio et al., 2023). Gavrikov and Keuper show the capacity of trainable combinations of random convolutions (Gavrikov & Keuper, 2023). These works make downstream adaptation an essential alternative explanation for poor frozen performance. Our fixed-feature classifier diagnostic is evidence of a training gap in the tested shallow models, not a claim that fixed filters generally fail or succeed. Nor do our small-model experiments establish a speed or accuracy improvement over those architectures.

Orozco-Solis et al. study parameterized Gabor filters on MNIST and Fashion-MNIST and propose using inter-filter correlation distance to identify when Gabor-layer training should stop (Orozco-Solis et al., 2024). Their other convolutional layers remain frozen during these experiments. This is a direct precedent for monitoring filter degradation during learning, but differs from releasing a soft template-retention penalty while continuing training. Their proposed explanation involving the fully connected layers is speculative; it does not establish the causal mechanism in our models.

2.2 What kernel appearance and concept measurements establish

Chowers and Weiss analyze first-layer energy profiles and connect their consistency, including under random labels, to image statistics (Chowers & Weiss, 2023). Jorgenson et al. examine how properties such as sharpness, noise and color leave signatures in early filters (Jorgenson et al., 2026). These results caution against treating recognizable kernel structure as sufficient evidence of semantic function. Our spectrum controls address some low-level properties, while renderer labels provide a distinct semantic measurement; neither removes every alternative explanation.

Network Dissection quantifies unit–concept correspondence (Bau et al., 2017), and Bau et al. subsequently examine causal roles of concept-related units through interventions (Bau et al., 2020). Concept Whitening explicitly aligns activation-space axes with known concepts (Chen et al., 2020). Our bank alignment instead concerns **weight space**, without concept supervision in the training objective. Thus high bank similarity, high object AUROC and foreground localization are different quantities. Their dissociation is an empirical result here, not a claim that the general distinction between representation and function is new. Related distinctions also arise in concept sidechannel models (Debot & Marra, 2025) and faithful concept-trace architectures (Parchami-Araghi et al., 2025), whose architectural assumptions and explanation mechanisms differ from ours.

2.3 Causal concepts and measurement-sensitive patching

CaCE distinguishes the effect of changing a concept from observational correlation (Goyal et al., 2020). Our renderer directly constructs matched concept changes, but U is not CaCE: it measures how well internal patches reproduce this particular model's probability response relative to random patches. Zhang and Nanda demonstrate that corruption, output metric and joint patching choices can change localization results in language models; their sliding-window analysis already highlights dependence on intervention granularity (Zhang & Nanda, 2024). Heimersheim and Nanda distinguish exploratory localization, verification, and the evidence supplied by different patching procedures (Heimersheim & Nanda, 2024). Our measurements should be read with those distinctions, not as identifying a complete circuit or proving necessity.

Miller et al. show that circuit faithfulness depends on ablation choices, including which components receive replacement activations and whether the circuit or its complement is intervened on (Miller et al., 2024). Our channel-map replacement changes all downstream uses of the selected activations; it does not isolate a circuit by ablating its complement. Stage C varies channel count at fixed channel granularity, and U is not their circuit-faithfulness score.

Nicolson et al. demonstrate layer inconsistency, concept entanglement and spatial dependence in CAV/TCAV analyses, including controlled shape–colour associations in Elements (Nicolson et al., 2025). Our renderer-based channel AUROC and foreground IoU are different measurements; independent annotations do not by themselves establish functional use. Sharma and Le compare demographic encoding with SAE-feature patching, ablation and steering in language models (Sharma & Le, 2026). Their comparison includes per-pair versus aggregate selection, multiple intervention sizes and variance-matched controls. We therefore claim neither the general encoding–influence distinction nor intervention-size sensitivity as new. Our training manipulation and retention–release comparison address a different empirical question.

The formal causal-abstraction literature also separates intervention agreement from unrestricted claims about explanations (Geiger et al., 2025). The non-identifiability analyses of Méloux et al. (Méloux et al., 2025) and alignment-map limitations studied by Sutter et al. (Sutter et al., 2025) concern different mathematical settings; our channel-patching experiment neither proves nor resolves those problems. We make no unique-mechanism claim.

2.4 What this paper adds relative to these precedents

Table 1: Relationship to the reviewed literature

Closest line of work	Established contribution	Specific role of the present study
Structured initialization and preservation (Özbulak & Ekenel, 2018; Molaei & Shiri Ahmad Abadi, 2020; Ma et al., 2020; Wang & Alkhalifah, 2024)	Insert task-related filter structure and preserve a parameterized family	Compare soft retention and release with independent concept measures and matched interventions
Fixed/combined spatial filters (Linse et al., 2023; Gaudio et al., 2023; Gavrikov & Keuper, 2023)	Learn useful predictors with fixed structured or random filters and downstream mixing	Diagnose fixed-feature readout limitations; do not equate a short-run deficit with representation capacity
First-layer statistics (Chowers & Weiss, 2023; Jorgenson et al., 2026)	Explain or infer low-level data properties from filter weights	Keep low-level appearance/spectrum separate from object concept and intervention outcomes
Unit concepts and interventions (Bau et al., 2017; 2020; Chen et al., 2020; Debot & Marra, 2025; Parchami-Araghi et al., 2025; Goyal et al., 2020)	Quantify semantics, build concept-aligned representations or examine causal concept/unit roles	Measure these quantities while manipulating a template prior rather than training a concept bottleneck
Patching methods (Zhang & Nanda, 2024; Heimersheim & Nanda, 2024; Miller et al., 2024)	Show methodological and granularity sensitivity of circuit localization	Show a retention-versus-release comparison reversing across channel budgets in small vision models

Table 1: Relationship to the reviewed literature

Closest line of work	Established contribution	Specific role of the present study
Causal abstraction and identifiability (Geiger et al., 2025; Méloux et al., 2025; Sutter et al., 2025)	Formalize causal explanations and their limitations	Restrict the claim to the measured finite interventions, without a general identification theorem

The reviewed representation–influence and concept-measurement precedents (Nicolson et al., 2025; Sharma & Le, 2026) further constrain novelty: this study does not introduce independent concept assessment or selected-versus-control patching. The distinguishing contribution is the **combination of intervention-controlled training comparisons and the observed budget-dependent reversal**, not any component in isolation. Relative to the reviewed sources, this is a specific empirical extension at the intersection of structured filters and patching evaluation. The comparison does not establish that no unreviewed paper contains the same experiment. The source-by-source evidence map records what was inspected and what each comparison can support.

3 Methods

3.1 Experimental stages and inferential roles

Table 2: Experimental stages and inferential roles

Stage	Training/evaluation scope	Data blocks	Statistical status
A: alignment and controls	400 models, 40 epochs, 10 conditions, 2 tasks, 2 architectures	0–9	Four locally prospective primary U tests; remaining comparisons exploratory
B: retention and release	200 models, 200 epochs, 5 conditions, 2 tasks, 2 architectures	2000–2009	Design saved before training; reported comparisons exploratory
C: measurement robustness	80 final checkpoints selected from B; 3 rankings × 4 sizes	Same as B	Post hoc reanalysis of existing models and test data

A historical 28-run pilot motivated foundation repairs and is archived for provenance, but is excluded from the paper's clean comparison tables. The 600 trained models in A and B are not 600 independent draws of one treatment effect: conditions share data and initialization components within blocks, and the two stages use different horizons and treatment sets. Stage C adds no independent training sample.

Stages A and B use the same corrected renderer, architectures and bank. Stage A includes random, random_unitnorm, frozen_random_unitnorm, template_init, template_retention_0.1, template_retention_1, frozen_templates, spectrum_init, spectrum_retention_1 and frozen_spectrum. A common Fourier phase transform of the template bank preserves each kernel's Fourier magnitude and the bank's Gram matrix, singular values and rank at initialization. Frozen controls preserve their anchors; trainable controls can drift. They are matched spectral controls, not guaranteed semantically empty filters. "Frozen" refers only to the first convolution; the second convolution remains trainable when present.

Stage A uses Adam at 0.003, batch 128 and 40 epochs (160 updates), final-checkpoint evaluation and training/validation-only selection rules. The main Stage B/C methods follow below. Full Stage A details and its locally timestamped protocol remain available.

3.2 Design, streams and data

In Stage B, the factorial design contains 2 tasks × 2 architectures × 10 blocks × 5 conditions = **200 runs**. Each model trains for 200 epochs. Blocks 2000–2009 use fresh deterministic streams, distinct from the earlier

study's blocks 0–9. The master seed is 2026090617. Within a block, conditions share data, minibatch ordering and downstream initialization; block differences include both generated-data and initialization variation. The unit for paired uncertainty estimates is the block, not an image, epoch or intervention pair.

Each task/block contains 512 training, 256 validation and 256 test images. These correspond to 128, 64 and 64 independently generated nuisance contexts, respectively, each rendered in all four balanced states. Images are 32×32 grayscale. Stored data include labels, renderer concept indicators, visible masks, matched nuisance variants and context metadata. All splits use separate seed namespaces. Each training stage contains 20,480 main images shared across model conditions, plus nuisance variants. Stage C reuses Stage B images.

single_shape classifies line, circle, triangle and square. Every context renders all four identities under the same nuisance realization. **two_concepts** renders two objects: square/circle and line/triangle. The label is circle_present + 2×triangle_present. Object positions are swapped independently across contexts; single-bit interventions alter one identity while retaining the other.

Rendering varies angle from 0 to $\pi/4$, width from 0.8 to 1.6, scale from 0.85 to 1.15, translation, and Gaussian noise with standard deviation sampled between 0 and 0.05. A 5×5 occluder occurs with probability 0.1. Threefold antialiasing is used. Matched counterfactual images share nuisance context; separate nuisance images change translation and noise while preserving identities and the other sampled settings. This controlled distribution is not a natural-image benchmark.

3.3 Models and priors

TinyCNN uses a 1→16 convolution with 9×9 kernels, padding 4 and no bias, followed by ReLU, global max pooling and a four-class linear head (1,364 parameters). **TwoLayerCNN** adds a trainable 16→16 convolution with 3×3 kernels and ReLU before pooling (3,668 parameters). Both are small architectures; the comparison isolates added depth within this family, not broad architectural generalization.

The corrected prior contains 8 edge, 4 corner and 4 ring filters. Kernels are centered and normalized to unit norm. The bank has rank 10: genuine two-ray corners correct a flaw in the original archive, but edge redundancies remain. Stage B does not include Stage A's spectrum controls, so it cannot isolate every geometric confound in this new schedule comparison.

Table 3: Initialization and retention schedules

Condition	Initialization	Retention schedule
random	Default PyTorch random	λ=0
random_unitnorm	Same random draw, centered and unit norm	λ=0
template_init	Corrected template bank	λ=0
template_retention_1	Corrected template bank	λ=1 throughout
template_release	Corrected template bank	λ=1 through epoch 10, decreasing linearly to 0 at epoch 80

All layers in Stage B are trainable; its five conditions do not include freezing. The release schedule is $\lambda(e)=\text{clip}((80\text{-}e)/70,0,1)$. Release and constant-retention histories match exactly through epoch 10 in the saved measurements. Early differences between those two conditions therefore cannot be evidence for a release effect.

3.4 Optimization and recording

The loss is cross-entropy plus $\lambda\|W - W_{\text{anchor}}\|_F^2/\|W_{\text{anchor}}\|_F^2$, applied only to the first convolution. All template anchors have squared norm 16. Adam uses learning rate 0.003 and batch size 128; four optimizer steps per epoch give 800 steps per model. There is no learning-rate scheduler, early stopping or checkpoint selection. The final-epoch outcome is reported as final performance; 200 epochs is not assumed to guarantee convergence.

Every epoch records sample-weighted training loss/accuracy, validation loss/accuracy, retention coefficient and penalty, kernel alignment/norm, first-layer gradient norm and pooled feature norm. Training metrics aggregate predictions made during parameter updates; validation metrics evaluate the completed epoch. Saved checkpoints occur at initialization and epochs 1, 5, 10, 20, 40, 80, 120, 160 and 200. Model, optimizer and random state are retained for resumption.

The recorded environment is Python 3.11.16, PyTorch 2.14.0+cu130, NumPy 2.4.6 and Linux/WSL2, with two CPU threads. Despite the CUDA-enabled package name, the provided runner keeps tensors and models on CPU. Cross-study numerical comparisons can also reflect environment differences; the paired comparisons within this run are the principal evidence.

3.5 Independent concept and causal measurements

Alignment averages, over the 16 first-layer kernels, the maximum signed centered cosine similarity to the corrected bank. Multiple learned kernels may select the same template.

Concept AUROC measures held-out identity/presence discrimination by a single first-layer unit per concept. Channel and sign are chosen using validation-only standardized positive–negative activation contrasts. **Localization IoU** separately selects the validation-best foreground channel, using each channel's validation 95th-percentile activation threshold, and evaluates visible renderer masks on concept-positive test images. Localization channels can differ from AUROC channels. Neither metric selects channels by template resemblance.

Matched patching copies selected first-layer activation maps from a counterfactual image into its matched base image and propagates through the unchanged network tail. All 12 directed identity substitutions per test context are used for single_shape (768 pairs); two_concepts uses the eight directed one-bit changes (512 pairs). Validation concept contrasts select channels. The primary displayed patch size is four channels, compared with eight fixed random rankings of the same size; sizes one and eight are also retained.

For base probability p_0, actual image-counterfactual probability p_1 and internally patched probability pS, fidelity is

$$F = 1 - \frac{\sum \|p_S - p_1\|_2^2}{\sum \|p_1 - p_0\|_2^2}, \qquad U = F_{selected,4} - \mathrm{mean}(F_{random,4}).$$

No-op fidelity is zero and full-channel fidelity is one when the denominator is nonzero. F is not a mediated fraction and is unbounded below. U measures a selected-set advantage over random sets, not all causal information in the representation. Ground-truth counterfactual accuracy is reported separately because fidelity can reproduce an incorrect model response. Alternative linear-head refits use fixed features, three regularization values and validation-loss selection; they do not replace main causal metrics. All 1,800 selected refits report successful solver termination, which is not a proof of global representation capacity.

3.6 Analysis and integrity

Prospective tests (Stage A). Before its main outcomes were inspected, the local protocol fixed template_retention_1 minus template_init in U at k4 for each of four task/architecture settings. Two-sided paired t tests use ten block differences; Holm adjustment covers that family of four. Displayed 95% t intervals are marginal, not Holm-adjusted simultaneous intervals. An alignment increase of at least 0.10 is a descriptive strong-manipulation threshold; an accuracy loss greater than five percentage points is flagged. The protocol was frozen locally, not externally preregistered, and was informed by the historical pilot. The four decisions remain inconclusive; later exploration does not revise them.

Exploratory comparisons (Stages B/C and secondary A). We analyze early mean validation accuracy over epochs 1–10, first epoch reaching 95% validation accuracy, final test outcomes and final-20-epoch validation-loss slopes. First threshold attainment can be transient and is not a sustained-convergence measure. All 200 runs reached the threshold, so these particular means have no non-attainment censoring.

Paired differences use ten blocks per setting. We provide marginal 95% Student-t intervals, assuming reasonably behaved block differences. For paired differences $d_1, \ldots, d_{10}$, these are mean(d) $\pm$ $t_{0.975,9}$ SD(d)/$\sqrt{10}$. A difference between two subgroup point estimates is not itself a tested architecture interaction. The release schedule and training design were saved before execution; the present contrasts, displays and statistical analysis were assembled after outcomes were available. They are **exploratory**, not a newly preregistered confirmatory test family. Intervals are not adjusted for the many outcomes and contrasts. We do not promote isolated intervals excluding zero into confirmatory claims. Curves display means $\pm$ standard deviations across blocks; epochs are not independent replicates.

The upload manifest verified 12,129 files with no mismatches, and core/experiment hashes matched the supplied source. The present analysis independently checked the published tables against all raw histories and scalar evaluation JSON files: 40,000 epoch records and 1,800 checkpoint evaluations. Stored full/no-op identities and defined fidelity denominators passed checks. This is artifact and numerical-consistency validation; it does not independently rerun every saved model's forward pass. Raw imported evidence is unchanged.

3.7 GPU robustness evaluation of existing checkpoints

To test the dependence of the original four-channel result on its measurement, we reevaluate the final checkpoints for template_release and template_retention_1: 2 tasks $\times$ 2 architectures $\times$ 10 blocks $\times$ 2 conditions = **80 checkpoint evaluations**. These are a subset of the existing 200 models, not new training runs or independent replications. Each checkpoint produces 3 ranking methods $\times$ 4 intervention sizes = 12 method/size records, for **960 records** in total. The validation and test splits are unchanged.

We compare three validation-only rankings:

- **contrast:** the original standardized positive–negative concept activation contrast.
- **auroc:** descending absolute deviation of validation concept AUROC from 0.5. Inverse discrimination can therefore rank highly.
- **validation_patch:** descending single-channel counterfactual fidelity, calculated separately for each destination identity or toggled concept on validation pairs. This uses the model's downstream behavior, unlike the two observational rankings. It ranks singleton effects; it does not optimize a joint channel subset.

Each ranking is evaluated at k=1, 2, 4 and 8 channels. Every size has eight same-size random controls, using the same original seed namespace and shared rankings across conditions within a task/block. Test outcomes are not used to select channels. Undefined validation denominators invalidate the intervention-selected U rather than supporting a conclusion; no such invalid selections occurred in the uploaded results. Selected fidelity, random fidelity, U, counterfactual accuracy, prediction agreement and correct-pair conditional accuracy are retained separately.

Inference and patching use CUDA float32 with mixed precision and TF32 disabled. One checkpoint is evaluated at a time. Actual full-channel and no-op forward passes check the implementation. The original contrast/k4 outcome is compared with its stored CPU value: the maximum absolute U difference across all 80 checkpoints is 1.29×10^{-7}, and ordinary accuracy is identical. These checks support numerical agreement for the existing measurement; they do not validate every possible causal interpretation.

The robustness protocol was written with knowledge of the original results. It is a post hoc sensitivity analysis on reused data. For each task, architecture, ranking and size, release-minus-constant differences are paired over ten blocks. Intervals remain marginal 95% t intervals, with no multiplicity correction. We display the full intervention-size curve rather than selecting a favorable size. There are 96 displayed robustness contrasts across four settings, three rankings, four sizes and two outcomes (U and counterfactual accuracy); all are retained in the supplement. The Stage B endpoint file contains all 160 exploratory contrasts defined in the post-outcome analysis, including unfavorable outcomes. No new confirmatory family or equivalence margin is retroactively assigned. Shared contexts, targets, rankings, epochs and methods do not increase the number of independent blocks.

3.8 Exploratory kernel matching on saved checkpoints

After the preceding analyses, we compared the first-layer weights of all 200 Stage B models at initialization and nine scheduled checkpoints through epoch 200 (2,000 checkpoint states). No additional models were trained and no new forward evaluations were performed. For centered, unit-norm kernels and templates, let $S_{ij} = \langle \widehat{W}_i, \widehat{T}_j \rangle$. The existing alignment is $A_{\text{nearest}} = 16^{-1} \sum_i \max_j S_{ij}$. We additionally compute

$$A_{\text{assignment}} = \frac{1}{16} \max_{\pi \in \mathfrak{S}_{16}} \sum_{i=1}^{16} S_{i,\pi(i)}, \qquad G = A_{\text{nearest}} - A_{\text{assignment}}.$$

A maximum-weight bipartite assignment finds the permutation; unlike row maxima, it cannot reuse a template. We retain signed similarity because polarity changes can affect ReLU activations. The associated normalized Euclidean distance is $d_{ij} = \sqrt{\max(0, 2 - 2S_{ij})}$; distances describe the cosine-optimal assignment rather than a separate distance-optimal matching. We also record distinct nearest templates, inter-kernel redundancy, raw norms and same-index drift from initialization. A matching gap is a property of this bank, not a measure of human readability or independent semantic coverage, especially given its rank-10 redundancy.

Quantitative summaries include all ten blocks, with sample SD. Illustrations use block 2000, the lowest planned block, for every condition. Kernel panels share a centered/unit-norm colour scale; row reordering is performed independently, so a column does not track a persistent channel. We join saved concept AUROC, localization, accuracy, U and counterfactual accuracy at the same checkpoint. Within-condition endpoint Spearman associations use ten blocks, are descriptive, and are not treated as causal or multiplicity-controlled evidence. This follow-up is exploratory and adds no independent training replication.

4 Results

4.1 Prospective alignment tests and the readout diagnostic

Stage A increases alignment strongly in all four primary comparisons, but establishes no corrected U improvement. Differences below are template_retention_1 minus template_init after 40 epochs, not release effects.

Table 4: Prospective alignment tests at 40 epochs

Setting	Δalignment	Δaccuracy (pp)	ΔU [marginal 95% interval]	Holm p
single_shape / TinyCNN	+0.268	-0.23	+0.0345 [+0.0018, +0.0672]	0.1633
single_shape / TwoLayerCNN	+0.172	-0.04	-0.0092 [-0.0513, +0.0329]	1.0000
two_concepts / TinyCNN	+0.289	-6.13	+0.0252 [-0.0306, +0.0811]	0.9995
two_concepts / TwoLayerCNN	+0.198	-0.39	-0.0074 [-0.0557, +0.0408]	1.0000

The first marginal interval excludes zero, but its two-sided test does not survive Holm adjustment. None of these outcomes establishes equivalence or the absence of a useful effect. The shallow compositional setting additionally loses 6.13 accuracy percentage points, exceeding the predeclared descriptive flag. These results motivate investigation of optimization and measurement rather than a universal negative conclusion.

At fixed frozen-template features, a validation-selected alternative linear head raises TinyCNN mean test accuracy from 83.40% to 99.77% on single_shape and from 52.30% to 81.80% on two_concepts. This demonstrates unused predictive capacity within the tested fixed representation/readout family under the original training procedure. It does not establish an information-theoretic ceiling. Of 400 selected refits, 398 report successful solver termination; the two exceptions remain recorded. A further 200 head-patching

evaluations were designed after inspecting partial outcomes and are exploratory, not additional independent trained models. Their results are retained in the Stage A report.

4.2 Early template advantages depend on the setting

Table 5: Mean validation accuracy over epochs 1–10

Task / model	random	random_unitnorm	template_init	release / constant retention
single_shape / TinyCNN	79.46%	80.09%	76.23%	75.69%
single_shape / TwoLayerCNN	79.29%	84.21%	87.55%	87.52%
two_concepts / TinyCNN	64.10%	61.21%	54.37%	52.37%
two_concepts / TwoLayerCNN	71.13%	74.41%	72.72%	71.90%

Values are mean validation accuracy averaged over epochs 1–10. Template_init minus random_unitnorm is +3.34 percentage points [1.25, 5.42] for single_shape / TwoLayerCNN, but -6.84 [-10.26, -3.43] for two_concepts / TinyCNN. The other marginal intervals include zero. Thus even the direction of the early difference depends on task and architecture. The normalized random control reduces the apparent benefit relative to default random.

Time-to-threshold and early area are not interchangeable. On single_shape / TinyCNN, template_init has a lower early average but reaches 95% at mean epoch 7.7, compared with 8.2 for random_unitnorm. On two_concepts / TinyCNN, release reaches 95% at epoch 46.5 on average, versus 66.8 for constant retention, 23.4 for template_init and 25.5 for random_unitnorm. Release improves the constrained model's threshold time, but does not make it the fastest learner.

4.3 Longer training removes much of the apparent shallow-model deficit

On two_concepts / TinyCNN, template_init rises from 97.66% test accuracy at epoch 40 to 99.30% at epoch 200, matching random_unitnorm's final 99.30%. Constant retention also improves substantially, from 92.54% to 98.01%. The initial short-budget deficit therefore did not establish that templates were stuck at a fixed representational ceiling. The constrained model still improves late: its mean final-20-epoch validation-loss slope is negative. Continued improvement does not guarantee eventual equality, but it precludes treating the observed training horizon as a proven asymptote.

4.4 Release improves compositional accuracy but sacrifices alignment

Table 6: Release-minus-retention contrasts across tasks and architectures

Setting	Constant retention accuracy	Release accuracy	Paired gain, pp [95% interval]	Alignment: constant → release
single_shape / TinyCNN	99.96%	100.00%	+0.04 [-0.05, 0.13]	.959 → .690
single_shape / TwoLayerCNN	99.92%	99.96%	+0.04 [-0.05, 0.13]	1.000 → .942
two_concepts / TinyCNN	98.01%	99.22%	+1.21 [0.71, 1.71]	.947 → .580
two_concepts / TwoLayerCNN	99.14%	99.53%	+0.39 [0.04, 0.74]	.999 → .907

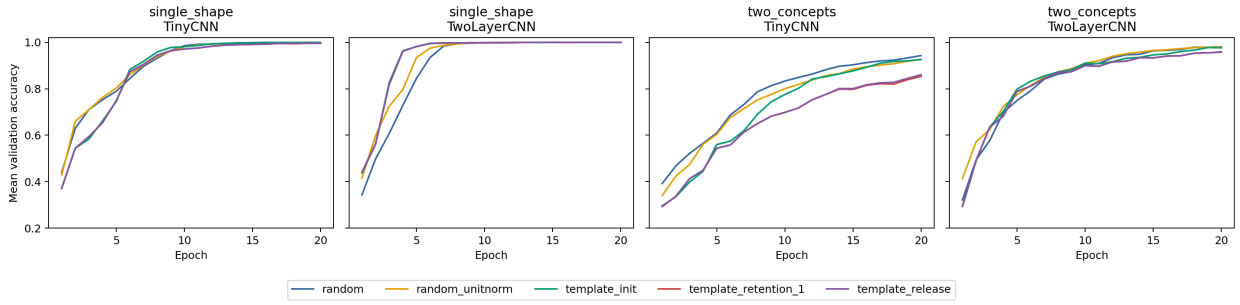

Figure 1: Mean validation accuracy over the first 20 epochs, by task, architecture and condition.

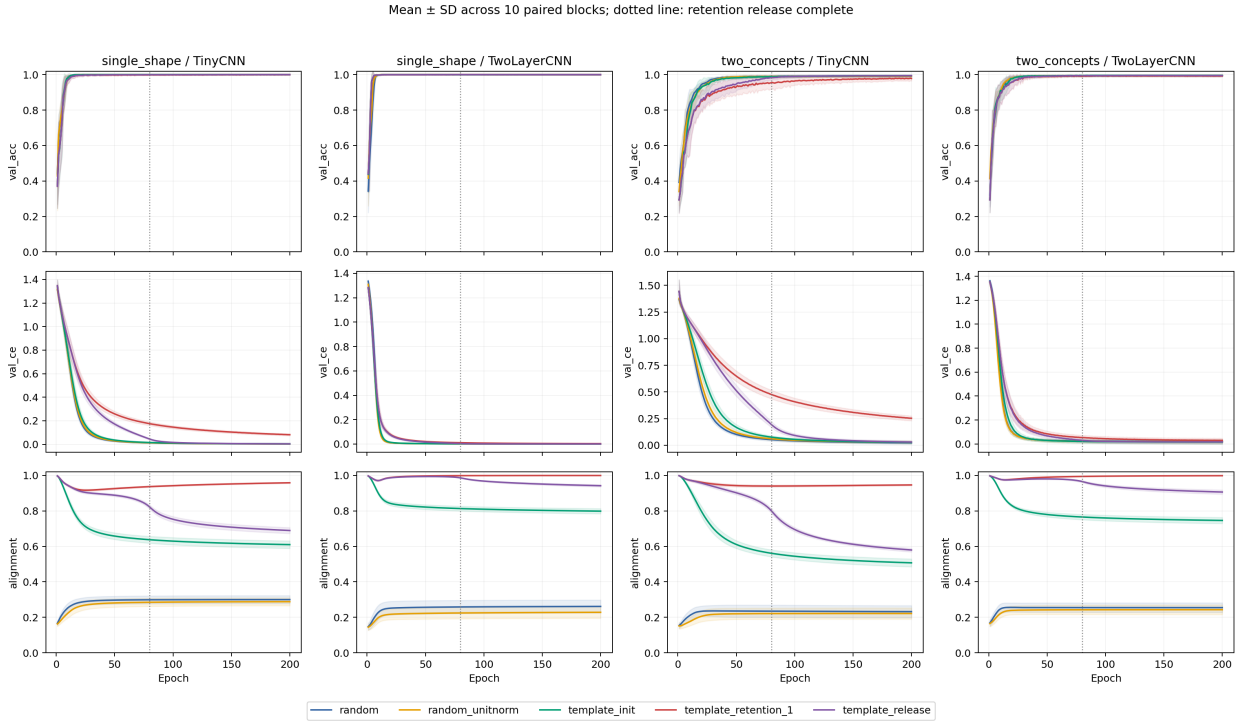

Figure 2: Learning and alignment trajectories over the complete training horizon; bands show block standard deviations.

Release reduces mean alignment in all four settings. It retains more alignment than template_init in each setting, indicating dependence on the optimization path after λ reaches zero. This does not isolate alignment as the cause of any accuracy difference; the schedule changes the learned representation and downstream adaptation jointly.

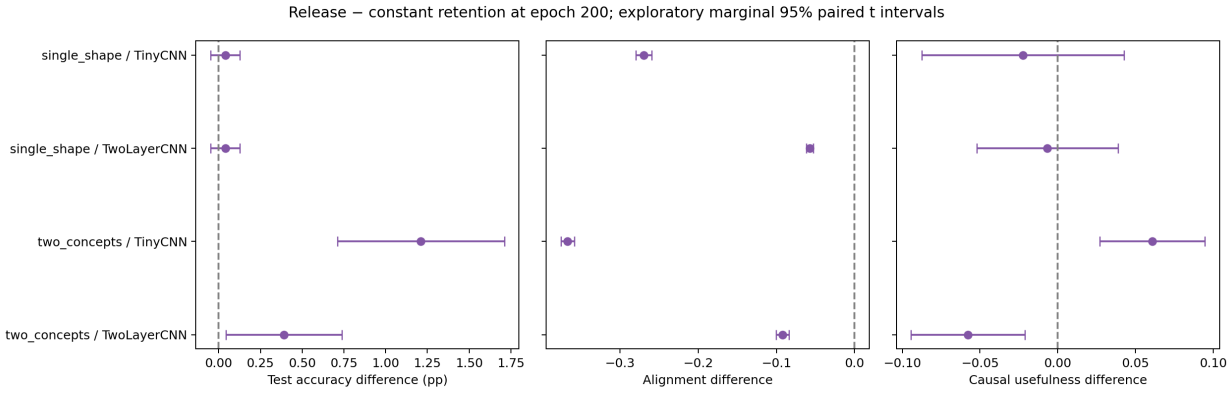

Figure 3: Paired final release-minus-constant effects with exploratory marginal 95% intervals.

4.5 Independent concept and causal outcomes do not move uniformly

Table 7: Concept and intervention measurements across tasks and architectures

Setting	AUROC constant → release	Localization IoU constant → release	U constant → release	Paired ΔU [95% interval]
single_shape / TinyCNN	.938 → .975	.544 → .438	.358 → .335	-.022 [-.087, .043]
single_shape / TwoLayerCNN	.874 → .846	.566 → .505	.067 → .060	-.007 [-.052, .039]
two_concepts / TinyCNN	.946 → .984	.347 → .235	.657 → .718	+.061 [.027, .095]
two_concepts / TwoLayerCNN	.776 → .771	.391 → .363	.061 → .003	-.058 [-.094, -.021]

In the compositional task, reduced retention is associated with higher predictive accuracy in both architectures, but opposite selected-channel U differences. In TinyCNN, reduced alignment accompanies better single-unit concept discrimination and a higher U; in TwoLayerCNN, the U advantage largely disappears. Localization falls in all four settings. Object foreground overlap, identity selectivity and selected-set patching usefulness therefore cannot be substituted for each other.

The distinction between relative fidelity and intervention correctness is visible even within TinyCNN: on two_concepts, release has U=.718 versus .657 for constant retention, while selected-patch counterfactual accuracy is **90.64% versus 93.61%**. A higher advantage over random patches need not mean higher absolute intervention accuracy. In TwoLayerCNN, selected counterfactual accuracy is only 4.02% for release despite 99.53% ordinary accuracy; four first-layer channels often fail to enact the intended change. This could reflect distributed coding, the channel-ranking rule or downstream nonlinear interaction, rather than absence of causal information.

4.6 The causal contrast depends on intervention size and selection

The GPU robustness evaluation changes the interpretation of the original architecture comparison. On two_concepts / TinyCNN, **all three selection methods** yield negative release-minus-constant U at k1 and k2, but positive differences at k4 and k8. This consistency across rankings makes a failure specific to the original contrast ranking an insufficient explanation. However, the reversal across sizes rules out describing release as uniformly more causally useful under these interventions.

The table gives mean paired ΔU [marginal 95% interval] on two_concepts. The corresponding results for both tasks, including absolute counterfactual accuracy, are retained in the complete paired table.

Table 8: Measurement sensitivity across channel budgets

Architecture	Ranking	k=1	k=2	k=4	k=8
TinyCNN	contrast	-0.335 [-0.425, -0.245]	-0.101 [-0.201, -0.001]	+0.061 [+0.027, +0.095]	+0.040 [+0.025, +0.055]
TinyCNN	auroc	-0.292 [-0.396, -0.189]	-0.125 [-0.223, -0.028]	+0.061 [+0.023, +0.098]	+0.040 [+0.026, +0.055]
TinyCNN	validation_patch	-0.237 [-0.283, -0.191]	-0.060 [-0.098, -0.021]	+0.067 [+0.044, +0.090]	+0.038 [+0.023, +0.054]
TwoLayerCNN	contrast	-0.001 [-0.008, +0.006]	-0.000 [-0.015, +0.015]	-0.058 [-0.094, -0.021]	-0.004 [-0.096, +0.089]
TwoLayerCNN	auroc	-0.003 [-0.010, +0.005]	+0.003 [-0.009, +0.016]	-0.054 [-0.092, -0.015]	+0.011 [-0.108, +0.130]

Table 8: Measurement sensitivity across channel budgets

Architecture	Ranking	k=1	k=2	k=4	k=8
TwoLayerCNN	validation_ patch	-0.013 [-0.019, -0.007]	-0.043 [-0.064, -0.023]	-0.021 [-0.095, +0.054]	+0.043 [+0.012, +0.074]

In TwoLayerCNN, contrast and AUROC retain negative k4 differences. Validation-patch ranking gives a negative k4 mean with an interval spanning zero, and a positive k8 difference. Thus the deeper-model conclusion also depends on the measurement. These marginal intervals describe the observed sensitivity; they are not separate confirmatory discoveries.

The figure shows relative U and absolute counterfactual-accuracy differences separately. U compares selected-channel reconstruction with a same-size random baseline, so its change can reflect either component. Because those baselines also change with k, a sign reversal in ΔU does not alone prove that causal information has become more distributed. It establishes that the comparative conclusion depends on intervention budget. Diagnosing concentration, redundancy or channel interactions requires additional analysis beyond ranking singleton effects.

4.7 One-to-one matching preserves the retained-kernel contrast

The recomputed nearest-template alignment agrees with all 1,800 saved post-training evaluations to the reported numerical precision (maximum absolute difference zero in this execution). At epoch 200, $A_{\text{assignment}}$ for constant retention ranges from 0.9468 to 0.9999 across the four settings, with zero matching gap to four decimal places. The release means range from 0.5743 to 0.9425; their gaps range from 0.0000 to 0.0054. Thus the high retained alignment is not explained by the reuse of the same nearest template in this comparison. For the random and normalized-random conditions, mean gaps range from 0.0241 to 0.0335 and mean distinct nearest-template counts from 8.3 to 9.7, compared with 16 for constant retention. These are bank-specific morphology measurements, not evidence of less useful random features. Appendix E reports all endpoint means and trajectories.

The change of matching rule does not erase the previously observed separation from behavioral measures. For example, on two_concepts/TinyCNN, constant retention has assignment similarity 0.9468 versus 0.5743 for release, while saved accuracy is 98.01% versus 99.22% and four-channel U is 0.657 versus 0.718. This comparison neither establishes that lowering similarity causes improvement nor makes the intervention ordering independent of channel budget. Figure 6 displays the underlying filters; no human readability experiment was performed.

5 Interpretation

The motivating early-advantage/late-constraint hypothesis receives partial, conditional support. Templates can provide an early advantage, as in single_shape / TwoLayerCNN, but do not generally do so. Template initialization alone catches up on the shallow compositional task. Constant retention leaves a residual fixed-budget cost, and release alleviates that cost. Thus initialization and persistent constraint must be distinguished before attributing late performance to templates themselves.

At the original four-channel measurement, release lowers alignment and improves compositional prediction in both architectures, while the selected-channel U difference changes sign with depth. The robustness evaluation shows that this is a conditional measurement result, not an established general mechanism: changing intervention size can reverse the shallow-model contrast, and changing selection and size alters deeper-model outcomes. The experiments support explicit separation of feature appearance, concept discrimination, localization, relative fidelity and absolute intervention correctness.

A plausible hypothesis is that retention changes the concentration or interaction of useful information across channels. That hypothesis remains unproven. Relative U includes a size-dependent random baseline; singleton rankings do not characterize joint effects, and patched activation combinations may be unnatural.

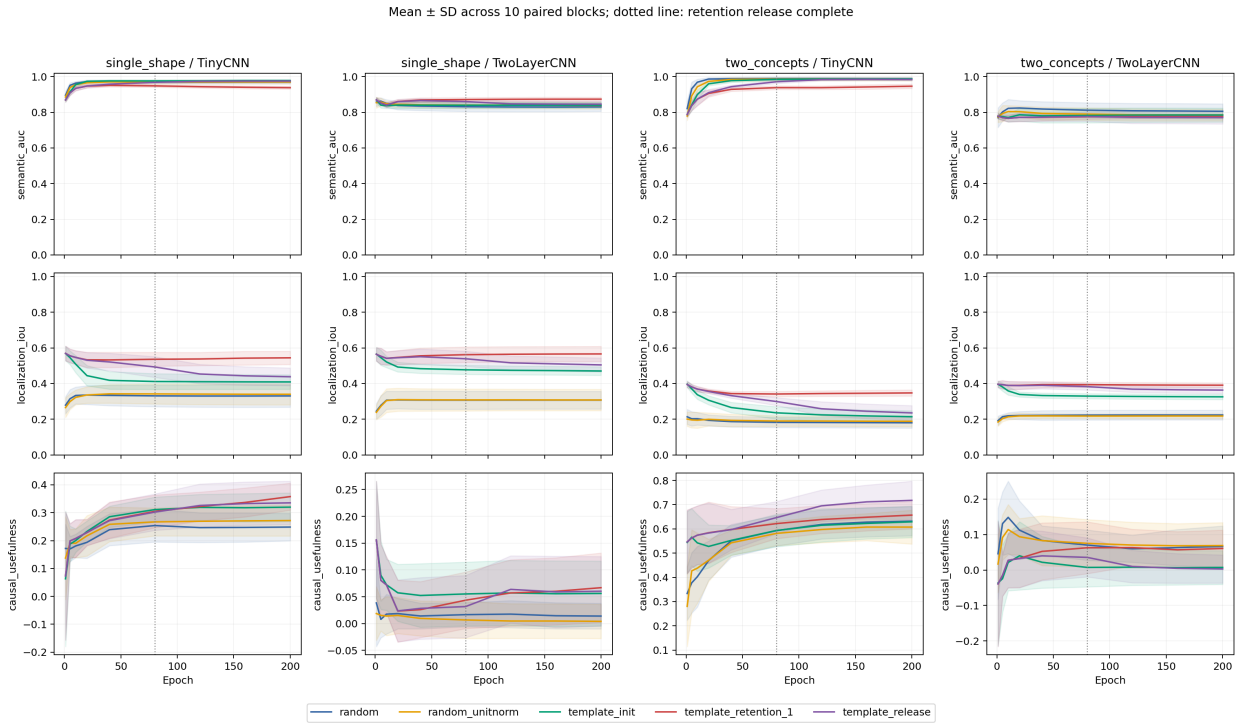

Figure 4: Concept and intervention measurements across recorded checkpoints; bands show block standard deviations.

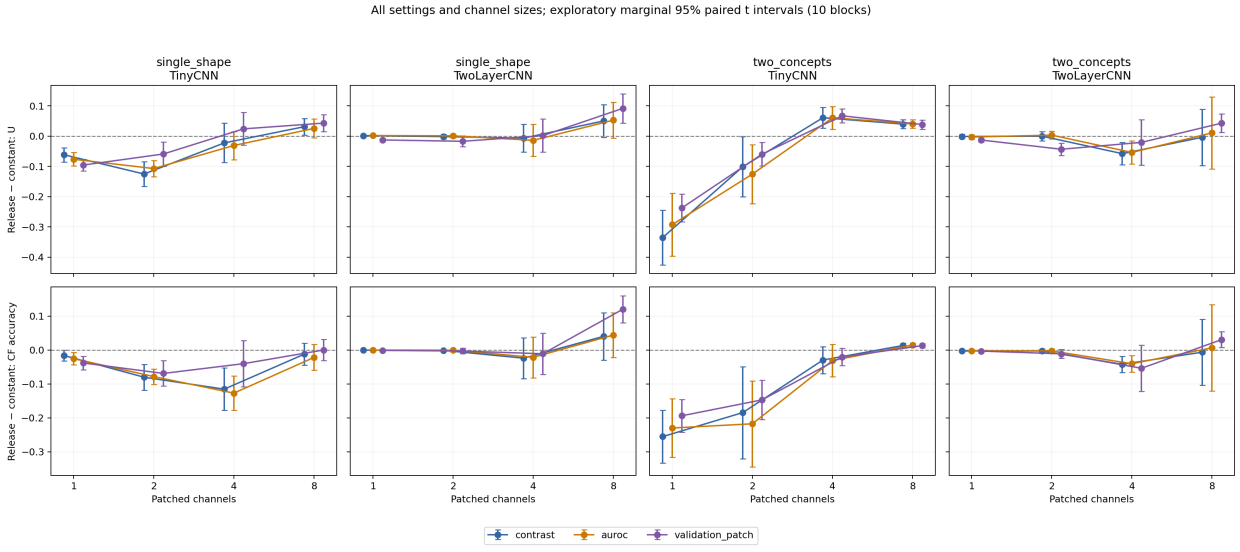

Figure 5: Release-minus-constant differences in U (top) and counterfactual accuracy (bottom), with identical vertical scales within each row. All four settings and all four sizes are shown. Intervals are marginal and exploratory; see Appendix C for all 96 numerical contrasts.

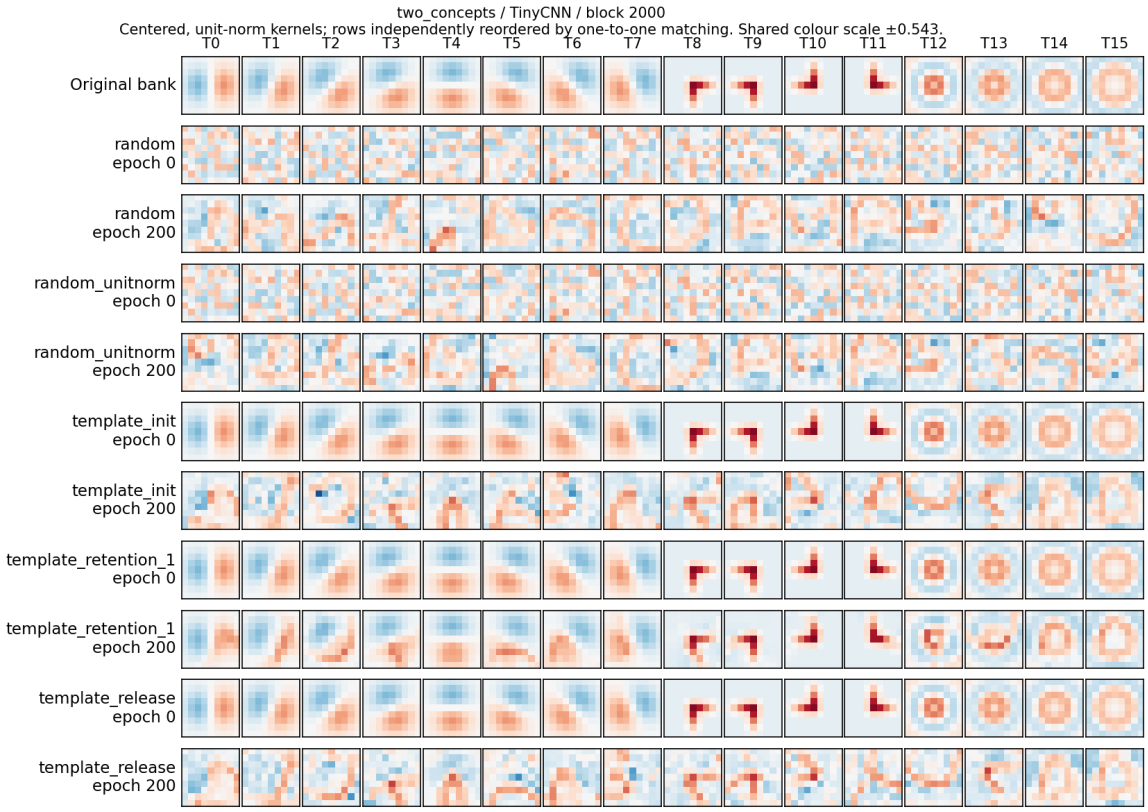

Figure 6: Original templates and saved first-layer kernels on two_concepts/TinyCNN, block 2000. Every condition is shown at epochs 0 and 200. Rows are independently ordered by maximum-weight one-to-one matching to the original bank; columns therefore indicate assigned templates, not persistent channel identity. Centering, unit normalization and a shared colour scale expose shape while removing raw amplitude. This fixed-block illustration is not a human interpretability test; numerical summaries use all blocks.

We therefore report measurement sensitivity as the observed result and distributed coding as a possible explanation requiring separate evidence.

These results do not supersede the earlier study's four inconclusive corrected tests. The release experiment uses fresh blocks, a longer horizon and a different treatment; the robustness analysis then reuses its final models and test data. Together they form one empirical investigation, with different evidential roles. Neither the number of checkpoints nor the number of ranking/size combinations should be presented as additional independent replication.

6 Limitations

- Only two related synthetic tasks, two small architectures and one template bank are covered. No natural-image or human-readability validation is present.
- Conditions share balanced rendered contexts. This improves control but simplifies real concept correlations. Whole-object concepts do not independently certify every local primitive represented in the prior.
- A single release schedule, optimizer and horizon were tested. No claim of optimal scheduling or convergence follows. There are only ten independent blocks per setting.
- Repeated checkpoint test evaluation was scheduled and does not alter training, but subsequent schedule development using these curves would require new held-out evidence.
- Exploration spans many outcomes, rankings, sizes and contrasts. Marginal intervals do not control familywise error, and no practical-equivalence margin was fixed. The robustness analysis reused the same test data after the original findings were known; validation-only selection does not make this an independent confirmation.
- Patching may generate unnatural combinations of feature maps even when source and target images are matched. Fidelity is tied to probability outputs, confidence and the chosen ranking/layer. Probability saturation and negative-component sensitivity are known issues (Zhang & Nanda, 2024; Heimersheim & Nanda, 2024); this study has not established robustness to logit- or divergence-based alternatives. Full/no-op identities are checks, not discoveries.
- Unit selection is refreshed from validation at each checkpoint. A metric trajectory can include changes in selected units, not just changes in one persistent feature. Stored probe arrays permit future unit-tracking analyses.
- The current result audit checks stored evidence rather than independently reproducing all model evaluations. Solver termination flags are not guarantees of representation optimality.
- Broader normalization/rank/spectrum controls from the earlier study were not crossed with the release schedule. Runtime comparisons are not analyzed.

Our random-channel baseline is not matched on activation variance or patch magnitude. Sharma and Le use controls addressing these alternatives in their SAE setting (Sharma & Le, 2026). Stage A initialization spectrum matching does not supply such an intervention control. Whether the retention–release ordering survives that comparison remains untested.

7 Scope and future work

The present evidence supports a scoped empirical account of learning dynamics and measurement sensitivity in small CNNs. It does not justify automatically launching confirmation of a broad "release helps shallow and harms deeper causal usefulness" claim, because that claim changes with the measurement.

A future independent study could test whether retention changes causal-effect concentration across channel budgets. Its protocol should specify the full k curve, an architecture-by-treatment-by-budget contrast, absolute counterfactual accuracy alongside U, and a multiplicity plan before generating outcomes. Additional annotated tasks and readout controls would test external relevance and possible mechanisms. These are future directions, not results of this paper. The final independent reproducibility review is deferred. The

current manuscript should not be described as fully submission-verified on the basis of stored-artifact checks alone.

8 Conclusion

Template initialization does not uniformly accelerate learning, and a short-budget deficit does not establish a template representation ceiling. Longer training lets the shallow compositional template-initialized model match normalization-matched random accuracy. Releasing retention improves on a constant constraint while sacrificing some template alignment, but its apparent causal-usefulness advantage depends on architecture, channel selection and intervention size. Across three rankings on two_concepts, the shallow-model contrast reverses between small and larger patches. These findings demonstrate why kernel appearance and a single patching score cannot substitute for a fully specified account of concept and intervention behavior. The study characterizes these dependencies while keeping kernel appearance, concept measurements and intervention outcomes distinct. One-to-one kernel matching preserves the retained-kernel contrast, but does not establish human interpretability or remove the dependence on intervention size.

9 Data and code availability

The experiments use procedurally rendered synthetic images. The research artifacts include the renderer, training and evaluation code, saved checkpoints, per-run metrics, and scripts that generate the tables and figures. Appendix A reports all primary contrasts; the subsequent appendices report endpoint condition means, the complete measurement-robustness comparison family and kernel-matching results.

The current working repository is not linked in this anonymous manuscript because it identifies the authors. An anonymized code and checkpoint archive has not yet been assembled. No archival DOI is assigned to the research artifacts, and the independent checkpoint-based reproducibility review remains outstanding.

References

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network Dissection: Quantifying Interpretability of Deep Visual Representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Bau_Network_Dissection_Quantifying_CVPR_2017_paper.html.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the Role of Individual Units in a Deep Neural Network. *Proceedings of the National Academy of Sciences*, 2020. doi: 10.1073/pnas.1907375117. URL https://doi.org/10.1073/pnas.1907375117.

Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept Whitening for Interpretable Image Recognition. *Nature Machine Intelligence*, 2020. doi: 10.1038/s42256-020-00265-z. URL https://doi.org/10.1038/s42256-020-00265-z.

Rhea Chowers and Yair Weiss. What do CNNs Learn in the First Layer and Why? A Linear Systems Perspective. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6115–6139, 2023. URL https://proceedings.mlr.press/v202/chowers23a.html.

David Debot and Giuseppe Marra. Quantifying the Accuracy-Interpretability Trade-Off in Concept-Based Sidechannel Models. arXiv preprint, 2025. URL https://arxiv.org/abs/2510.05670v2.

Alex Gaudio, Christos Faloutsos, Asim Smailagic, Pedro Costa, and Aurélio Campilho. ExplainFix: Explainable Spatially Fixed Deep Networks. *WIREs Data Mining and Knowledge Discovery*, 13(2):e1483, 2023. doi: 10.1002/widm.1483. URL https://doi.org/10.1002/widm.1483.

Paul Gavrikov and Janis Keuper. The Power of Linear Combinations: Learning with Random Convolutions. arXiv preprint, 2023. URL https://arxiv.org/abs/2301.11360v2.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal Abstraction: A Theoretical Foundation for Mechanistic Interpretability. *Journal of Machine Learning Research*, 26(83): 1–64, 2025. URL https://www.jmlr.org/papers/v26/23-0058.html.

Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. Explaining Classifiers with Causal Concept Effect (CaCE). arXiv preprint, 2020. URL https://arxiv.org/abs/1907.07165v2.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. arXiv preprint, 2024. URL https://arxiv.org/abs/2404.15255v1.

Grayson Jorgenson, Cassie Heine, Robin Cosbey, Abby Reynolds, Davis Brown, Henry Kvinge, Timothy Doster, and Tegan Emerson. Scratching the Surface: Reflections of Training Data Properties in Early CNN Filters. In *Proceedings of the 1st Conference on Topology, Algebra, and Geometry in Data Science (TAG-DS 2025)*, volume 321 of *Proceedings of Machine Learning Research*, pp. 166–175, 2026. URL https://proceedings.mlr.press/v321/jorgenson26a.html.

Christoph Linse, Erhardt Barth, and Thomas Martinetz. Convolutional Neural Networks Do Work with Pre-Defined Filters. In *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023. doi: 10.1109/IJCNN54540.2023.10191449. URL https://arxiv.org/abs/2411.18388v1.

Yangling Ma, Yixin Luo, and Zhouwang Yang. PCFNet: Deep Neural Network with Predefined Convolutional Filters. *Neurocomputing*, 382:32–39, 2020. doi: 10.1016/j.neucom.2019.11.075. URL https://doi.org/10.1016/j.neucom.2019.11.075.

Maxime Méloux, François Portet, Silviu Maniu, and Maxime Peyrard. Everything, Everywhere, All at Once: Is Mechanistic Interpretability Identifiable? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2502.20914v1.

Joseph Miller, Bilal Chughtai, and William Saunders. Transformer Circuit Faithfulness Metrics Are Not Robust. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=zSf8PJyQb2.

Somayeh Molaei and Mohammad Ebrahim Shiri Ahmad Abadi. Maintaining filter structure: A Gabor-based convolutional neural network for image analysis. *Applied Soft Computing*, 88:105960, 2020. doi: 10.1016/j.asoc.2019.105960. URL https://doi.org/10.1016/j.asoc.2019.105960.

Angus Nicolson, Lisa Schut, Alison J. Noble, and Yarin Gal. Explaining Explainability: Recommendations for Effective Use of Concept Activation Vectors. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=7CUluLpLxV.

Carlos Orozco-Solis, Alfonso Rojas-Domínguez, Héctor Puga, Manuel Ornelas Rodríguez, Martín Carpio, and Valentín Calzada-Ledesma. Learnable Gabor Filters in CNNs: Avoiding Filter Degeneration via Early Stopping Based on Similarity Metrics. In *Pattern Recognition*, volume 14755 of *Lecture Notes in Computer Science*, pp. 387–396, 2024. doi: 10.1007/978-3-031-62836-8_36. URL https://doi.org/10.1007/978-3-031-62836-8_36.

Gökhan Özbulak and Hazım Kemal Ekenel. Initialization of Convolutional Neural Networks by Gabor Filters. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2018. doi: 10.1109/SIU.2018.8404757. URL https://doi.org/10.1109/SIU.2018.8404757.

Amin Parchami-Araghi, Sukrut Rao, Jonas Fischer, and Bernt Schiele. FaCT: Faithful Concept Traces for Explaining Neural Network Decisions. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2510.25512v2.

Aarushi Sharma and Phong Le. Encoding Without Influence: Dissociating Demographic Representation from Causal Effect in Large Language Models. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=TQbXHsI3Lm.

Denis Sutter, Julian Minder, Thomas Hofmann, and Tiago Pimentel. The Non-Linear Representation Dilemma: Is Causal Abstraction Enough for Mechanistic Interpretability? In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2507.08802v2.

Fu Wang and Tariq Alkhalifah. Learnable Gabor kernels in convolutional neural networks for seismic interpretation tasks. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–9, 2024. doi: 10.1109/TGRS.2024.3365910. URL https://doi.org/10.1109/TGRS.2024.3365910.

Fred Zhang and Neel Nanda. Towards Best Practices of Activation Patching in Language Models: Metrics and Methods. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.16042v2.

A All prospective primary contrasts

Table 9: Complete prospective primary contrasts

Setting	ΔU [marginal 95% CI]	Raw p	Holm p	Δalignment	Δaccuracy (pp)
single_shape / TinyCNN	+0.0345 [+0.0018, +0.0672]	0.0408	0.1633	+0.268	-0.23
single_shape / TwoLayer-CNN	-0.0092 [-0.0513, +0.0329]	0.6337	1.0000	+0.172	-0.04
two_concepts / TinyCNN	+0.0252 [-0.0306, +0.0811]	0.3332	0.9995	+0.289	-6.13
two_concepts / TwoLayer-CNN	-0.0074 [-0.0557, +0.0408]	0.7358	1.0000	+0.198	-0.39

B All 200-epoch condition means

Mean ± sample SD across ten blocks. All five conditions and all four settings are shown. Accuracy is a percentage; U, alignment, AUROC and IoU are dimensionless.

Table 10: All endpoint condition means at 200 epochs

Setting	Condition	Accuracy (%)	Alignment	Concept AUROC	Localization IoU	U	Selected CF accuracy (%)
single_shape / TinyCNN	random	100.000 ± 0.000	0.300 ± 0.023	0.970 ± 0.009	0.330 ± 0.063	0.249 ± 0.049	27.174 ± 6.495
single_shape / TinyCNN	random_unitnorm	100.000 ± 0.000	0.288 ± 0.022	0.971 ± 0.011	0.340 ± 0.059	0.272 ± 0.055	30.716 ± 6.203
single_shape / TinyCNN	template_init	100.000 ± 0.000	0.610 ± 0.021	0.978 ± 0.006	0.409 ± 0.041	0.320 ± 0.051	35.469 ± 4.792
single_shape / TinyCNN	template_release	100.000 ± 0.000	0.690 ± 0.017	0.975 ± 0.007	0.438 ± 0.050	0.335 ± 0.078	38.620 ± 8.816
single_shape / TinyCNN	template_retention_1	99.961 ± 0.124	0.959 ± 0.003	0.938 ± 0.008	0.544 ± 0.037	0.358 ± 0.048	50.169 ± 5.142
single_shape / TwoLayerCNN	random	100.000 ± 0.000	0.261 ± 0.037	0.830 ± 0.025	0.307 ± 0.050	0.014 ± 0.024	2.799 ± 2.644
single_shape / TwoLayerCNN	random_unitnorm	100.000 ± 0.000	0.228 ± 0.033	0.838 ± 0.019	0.308 ± 0.060	0.004 ± 0.032	2.279 ± 1.985
single_shape / TwoLayerCNN	template_init	100.000 ± 0.000	0.799 ± 0.015	0.840 ± 0.016	0.470 ± 0.024	0.056 ± 0.061	5.924 ± 6.345
single_shape / TwoLayerCNN	template_release	99.961 ± 0.124	0.942 ± 0.006	0.846 ± 0.016	0.504 ± 0.039	0.060 ± 0.065	7.552 ± 7.152
single_shape / TwoLayerCNN	template_retention_1	99.922 ± 0.165	1.000 ± 0.000	0.874 ± 0.011	0.566 ± 0.043	0.067 ± 0.065	9.961 ± 10.034
two_concepts / TinyCNN	random	99.258 ± 0.892	0.232 ± 0.034	0.988 ± 0.009	0.180 ± 0.030	0.632 ± 0.060	73.906 ± 9.590
two_concepts / TinyCNN	random_unitnorm	99.297 ± 0.708	0.222 ± 0.031	0.986 ± 0.009	0.188 ± 0.030	0.607 ± 0.070	71.797 ± 10.205
two_concepts / TinyCNN	template_init	99.297 ± 0.514	0.507 ± 0.022	0.987 ± 0.005	0.214 ± 0.034	0.629 ± 0.065	77.949 ± 10.461
two_concepts / TinyCNN	template_release	99.219 ± 0.689	0.580 ± 0.011	0.984 ± 0.006	0.235 ± 0.041	0.718 ± 0.079	90.645 ± 6.945
two_concepts / TinyCNN	template_retention_1	98.008 ± 0.982	0.947 ± 0.004	0.946 ± 0.012	0.347 ± 0.017	0.657 ± 0.056	93.613 ± 3.604
two_concepts / TwoLayerCNN	random	99.492 ± 0.522	0.255 ± 0.027	0.806 ± 0.042	0.223 ± 0.027	0.066 ± 0.058	8.145 ± 6.561
two_concepts / TwoLayerCNN	random_unitnorm	99.375 ± 0.906	0.243 ± 0.028	0.785 ± 0.033	0.218 ± 0.013	0.069 ± 0.065	8.438 ± 5.469
two_concepts / TwoLayerCNN	template_init	99.570 ± 0.430	0.746 ± 0.017	0.785 ± 0.039	0.326 ± 0.015	0.007 ± 0.048	4.453 ± 4.286
two_concepts / TwoLayerCNN	template_release	99.531 ± 0.546	0.907 ± 0.012	0.771 ± 0.036	0.363 ± 0.018	0.003 ± 0.040	4.023 ± 3.412
two_concepts / TwoLayerCNN	template_retention_1	99.141 ± 0.576	0.999 ± 0.000	0.776 ± 0.010	0.391 ± 0.014	0.061 ± 0.052	8.242 ± 3.847

C Complete measurement-robustness contrasts

Release minus constant retention. All 96 contrasts: 4 settings × 3 rankings × 4 sizes × 2 outcomes. Intervals are marginal 95% paired t intervals; no multiplicity adjustment. Neither exclusion of zero nor agreement of several dependent intervals is a confirmatory discovery. CF accuracy differences are shown as proportions.

C.1 causal_usefulness

Table 11: Complete robustness contrasts for causal usefulness

Setting	Ranking	k	Mean difference	95% interval
single_shape / TinyCNN	auroc	1	-0.0764	[-0.0984, -0.0544]
single_shape / TinyCNN	auroc	2	-0.1071	[-0.1337, -0.0805]
single_shape / TinyCNN	auroc	4	-0.0309	[-0.0774, +0.0155]
single_shape / TinyCNN	auroc	8	+0.0255	[-0.0057, +0.0567]
single_shape / TinyCNN	contrast	1	-0.0616	[-0.0852, -0.0380]
single_shape / TinyCNN	contrast	2	-0.1252	[-0.1661, -0.0843]
single_shape / TinyCNN	contrast	4	-0.0224	[-0.0874, +0.0426]
single_shape / TinyCNN	contrast	8	+0.0312	[+0.0042, +0.0583]
single_shape / TinyCNN	validation_patch	1	-0.0955	[-0.1144, -0.0767]
single_shape / TinyCNN	validation_patch	2	-0.0590	[-0.0991, -0.0189]
single_shape / TinyCNN	validation_patch	4	+0.0240	[-0.0301, +0.0782]
single_shape / TinyCNN	validation_patch	8	+0.0430	[+0.0156, +0.0704]
single_shape / TwoLayerCNN	auroc	1	+0.0019	[-0.0001, +0.0039]
single_shape / TwoLayerCNN	auroc	2	+0.0011	[-0.0054, +0.0076]
single_shape / TwoLayerCNN	auroc	4	-0.0139	[-0.0668, +0.0391]
single_shape / TwoLayerCNN	auroc	8	+0.0528	[-0.0067, +0.1123]
single_shape / TwoLayerCNN	contrast	1	+0.0016	[-0.0003, +0.0035]
single_shape / TwoLayerCNN	contrast	2	-0.0015	[-0.0090, +0.0060]
single_shape / TwoLayerCNN	contrast	4	-0.0066	[-0.0521, +0.0388]
single_shape / TwoLayerCNN	contrast	8	+0.0503	[-0.0042, +0.1047]
single_shape / TwoLayerCNN	validation_patch	1	-0.0126	[-0.0182, -0.0070]
single_shape / TwoLayerCNN	validation_patch	2	-0.0170	[-0.0346, +0.0005]
single_shape / TwoLayerCNN	validation_patch	4	+0.0017	[-0.0530, +0.0564]

Table 11: Complete robustness contrasts for causal usefulness

Setting	Ranking	k	Mean difference	95% interval
single_shape / TwoLayerCNN	validation_patch	8	+0.0920	[+0.0437, +0.1402]
two_concepts / TinyCNN	auroc	1	-0.2923	[-0.3960, -0.1886]
two_concepts / TinyCNN	auroc	2	-0.1254	[-0.2228, -0.0279]
two_concepts / TinyCNN	auroc	4	+0.0606	[+0.0230, +0.0982]
two_concepts / TinyCNN	auroc	8	+0.0405	[+0.0259, +0.0551]
two_concepts / TinyCNN	contrast	1	-0.3352	[-0.4250, -0.2453]
two_concepts / TinyCNN	contrast	2	-0.1011	[-0.2007, -0.0015]
two_concepts / TinyCNN	contrast	4	+0.0608	[+0.0270, +0.0947]
two_concepts / TinyCNN	contrast	8	+0.0398	[+0.0248, +0.0548]
two_concepts / TinyCNN	validation_patch	1	-0.2371	[-0.2832, -0.1909]
two_concepts / TinyCNN	validation_patch	2	-0.0595	[-0.0978, -0.0213]
two_concepts / TinyCNN	validation_patch	4	+0.0672	[+0.0440, +0.0903]
two_concepts / TinyCNN	validation_patch	8	+0.0383	[+0.0229, +0.0538]
two_concepts / TwoLayerCNN	auroc	1	-0.0028	[-0.0101, +0.0045]
two_concepts / TwoLayerCNN	auroc	2	+0.0034	[-0.0093, +0.0160]
two_concepts / TwoLayerCNN	auroc	4	-0.0539	[-0.0924, -0.0153]
two_concepts / TwoLayerCNN	auroc	8	+0.0109	[-0.1079, +0.1297]
two_concepts / TwoLayerCNN	contrast	1	-0.0010	[-0.0084, +0.0063]
two_concepts / TwoLayerCNN	contrast	2	-0.0000	[-0.0152, +0.0152]
two_concepts / TwoLayerCNN	contrast	4	-0.0578	[-0.0944, -0.0212]
two_concepts / TwoLayerCNN	contrast	8	-0.0038	[-0.0964, +0.0889]
two_concepts / TwoLayerCNN	validation_patch	1	-0.0129	[-0.0191, -0.0066]
two_concepts / TwoLayerCNN	validation_patch	2	-0.0433	[-0.0640, -0.0227]
two_concepts / TwoLayerCNN	validation_patch	4	-0.0206	[-0.0954, +0.0542]
two_concepts / TwoLayerCNN	validation_patch	8	+0.0429	[+0.0123, +0.0735]

C.2 cf_accuracy

Table 12: Complete robustness contrasts for counterfactual accuracy

Setting	Ranking	k	Mean difference	95% interval
single_shape / TinyCNN	auroc	1	-0.0255	[-0.0438, -0.0073]
single_shape / TinyCNN	auroc	2	-0.0784	[-0.1011, -0.0557]
single_shape / TinyCNN	auroc	4	-0.1272	[-0.1782, -0.0762]
single_shape / TinyCNN	auroc	8	-0.0220	[-0.0601, +0.0161]
single_shape / TinyCNN	contrast	1	-0.0163	[-0.0322, -0.0003]
single_shape / TinyCNN	contrast	2	-0.0803	[-0.1181, -0.0426]
single_shape / TinyCNN	contrast	4	-0.1155	[-0.1782, -0.0528]
single_shape / TinyCNN	contrast	8	-0.0122	[-0.0442, +0.0197]
single_shape / TinyCNN	validation_patch	1	-0.0383	[-0.0583, -0.0183]
single_shape / TinyCNN	validation_patch	2	-0.0691	[-0.1066, -0.0317]
single_shape / TinyCNN	validation_patch	4	-0.0404	[-0.1083, +0.0275]
single_shape / TinyCNN	validation_patch	8	-0.0000	[-0.0312, +0.0312]
single_shape / TwoLayerCNN	auroc	1	-0.0005	[-0.0017, +0.0007]
single_shape / TwoLayerCNN	auroc	2	-0.0007	[-0.0019, +0.0006]
single_shape / TwoLayerCNN	auroc	4	-0.0216	[-0.0818, +0.0386]
single_shape / TwoLayerCNN	auroc	8	+0.0439	[-0.0217, +0.1095]
single_shape / TwoLayerCNN	contrast	1	-0.0005	[-0.0017, +0.0007]
single_shape / TwoLayerCNN	contrast	2	-0.0012	[-0.0027, +0.0004]
single_shape / TwoLayerCNN	contrast	4	-0.0241	[-0.0841, +0.0359]
single_shape / TwoLayerCNN	contrast	8	+0.0402	[-0.0295, +0.1099]
single_shape / TwoLayerCNN	validation_patch	1	-0.0012	[-0.0032, +0.0009]
single_shape / TwoLayerCNN	validation_patch	2	-0.0031	[-0.0109, +0.0047]
single_shape / TwoLayerCNN	validation_patch	4	-0.0112	[-0.0721, +0.0497]
single_shape / TwoLayerCNN	validation_patch	8	+0.1203	[+0.0805, +0.1602]
two_concepts / TinyCNN	auroc	1	-0.2301	[-0.3162, -0.1440]
two_concepts / TinyCNN	auroc	2	-0.2176	[-0.3442, -0.0910]
two_concepts / TinyCNN	auroc	4	-0.0311	[-0.0790, +0.0169]

Table 12: Complete robustness contrasts for counterfactual accuracy

Setting	Ranking	k	Mean difference	95% interval
two_concepts / TinyCNN	auroc	8	+0.0146	[+0.0087, +0.0205]
two_concepts / TinyCNN	contrast	1	-0.2555	[-0.3332, -0.1777]
two_concepts / TinyCNN	contrast	2	-0.1846	[-0.3204, -0.0487]
two_concepts / TinyCNN	contrast	4	-0.0297	[-0.0695, +0.0101]
two_concepts / TinyCNN	contrast	8	+0.0133	[+0.0070, +0.0195]
two_concepts / TinyCNN	validation_patch	1	-0.1939	[-0.2421, -0.1458]
two_concepts / TinyCNN	validation_patch	2	-0.1471	[-0.2049, -0.0892]
two_concepts / TinyCNN	validation_patch	4	-0.0205	[-0.0461, +0.0051]
two_concepts / TinyCNN	validation_patch	8	+0.0129	[+0.0068, +0.0190]
two_concepts / TwoLayerCNN	auroc	1	-0.0027	[-0.0062, +0.0008]
two_concepts / TwoLayerCNN	auroc	2	-0.0027	[-0.0085, +0.0031]
two_concepts / TwoLayerCNN	auroc	4	-0.0406	[-0.0652, -0.0161]
two_concepts / TwoLayerCNN	auroc	8	+0.0064	[-0.1208, +0.1337]
two_concepts / TwoLayerCNN	contrast	1	-0.0029	[-0.0064, +0.0005]
two_concepts / TwoLayerCNN	contrast	2	-0.0031	[-0.0096, +0.0033]
two_concepts / TwoLayerCNN	contrast	4	-0.0422	[-0.0659, -0.0185]
two_concepts / TwoLayerCNN	contrast	8	-0.0064	[-0.1039, +0.0910]
two_concepts / TwoLayerCNN	validation_patch	1	-0.0033	[-0.0066, -0.0001]
two_concepts / TwoLayerCNN	validation_patch	2	-0.0113	[-0.0247, +0.0020]
two_concepts / TwoLayerCNN	validation_patch	4	-0.0539	[-0.1222, +0.0144]
two_concepts / TwoLayerCNN	validation_patch	8	+0.0305	[+0.0073, +0.0536]

D Data and analysis map

- Prospective raw per-run results: `studies/cnn_causal_milestone/analysis/per_run.csv`.
- Retention-release: `analysis/retention_release_001/endpoints.csv` and `paired_contrasts.csv` (all 160 exploratory endpoint contrasts).
- Robustness raw method/size data: `results/patch_robustness_gpu_001/analysis/metrics.csv`.
- Rebuild these tables and the full sensitivity figure with `python analysis/main_paper/build_assets.py`.
- This script checks completeness and presentation, not saved model forwards. The final independent reproducibility review is deferred.

E Complete kernel-matching follow-up

All values below are endpoint means ± sample SD across ten blocks. The nearest and one-to-one scores use signed cosines. Coverage counts nearest-template identities, not independent semantic concepts. The analysis uses only saved Stage B checkpoints; no spectrum or frozen conditions from Stage A were added. These descriptive comparisons are exploratory.

Table 13: All endpoint kernel-matching means.

Setting	Condition	Nearest	One-to-one	Coverage
single_shape / TinyCNN	random	0.300 ± 0.023	0.267 ± 0.033	9.700 ± 1.703
single_shape / TinyCNN	random_unitnorm	0.288 ± 0.022	0.256 ± 0.034	9.700 ± 1.703
single_shape / TinyCNN	template_init	0.610 ± 0.021	0.605 ± 0.021	13.300 ± 1.337
single_shape / TinyCNN	template_release	0.690 ± 0.017	0.686 ± 0.017	14.200 ± 1.033
single_shape / TinyCNN	template_retention_1	0.959 ± 0.003	0.959 ± 0.003	16.000 ± 0.000
single_shape / TwoLayerCNN	random	0.261 ± 0.037	0.230 ± 0.039	9.200 ± 1.317
single_shape / TwoLayerCNN	random_unitnorm	0.228 ± 0.033	0.195 ± 0.039	8.300 ± 1.889
single_shape / TwoLayerCNN	template_init	0.799 ± 0.015	0.799 ± 0.015	15.700 ± 0.483
single_shape / TwoLayerCNN	template_release	0.942 ± 0.006	0.942 ± 0.006	16.000 ± 0.000
single_shape / TwoLayerCNN	template_retention_1	1.000 ± 0.000	1.000 ± 0.000	16.000 ± 0.000
two_concepts / TinyCNN	random	0.232 ± 0.034	0.199 ± 0.035	9.100 ± 1.524
two_concepts / TinyCNN	random_unitnorm	0.222 ± 0.031	0.191 ± 0.037	9.200 ± 1.398
two_concepts / TinyCNN	template_init	0.507 ± 0.022	0.496 ± 0.022	12.000 ± 1.054
two_concepts / TinyCNN	template_release	0.580 ± 0.011	0.574 ± 0.010	12.900 ± 0.994
two_concepts / TinyCNN	template_retention_1	0.947 ± 0.004	0.947 ± 0.004	16.000 ± 0.000
two_concepts / TwoLayerCNN	random	0.255 ± 0.027	0.231 ± 0.024	9.600 ± 1.075
two_concepts / TwoLayerCNN	random_unitnorm	0.243 ± 0.028	0.218 ± 0.027	9.000 ± 1.054
two_concepts / TwoLayerCNN	template_init	0.746 ± 0.017	0.746 ± 0.017	15.400 ± 0.699
two_concepts / TwoLayerCNN	template_release	0.907 ± 0.012	0.907 ± 0.012	16.000 ± 0.000
two_concepts / TwoLayerCNN	template_retention_1	0.999 ± 0.000	0.999 ± 0.000	16.000 ± 0.000

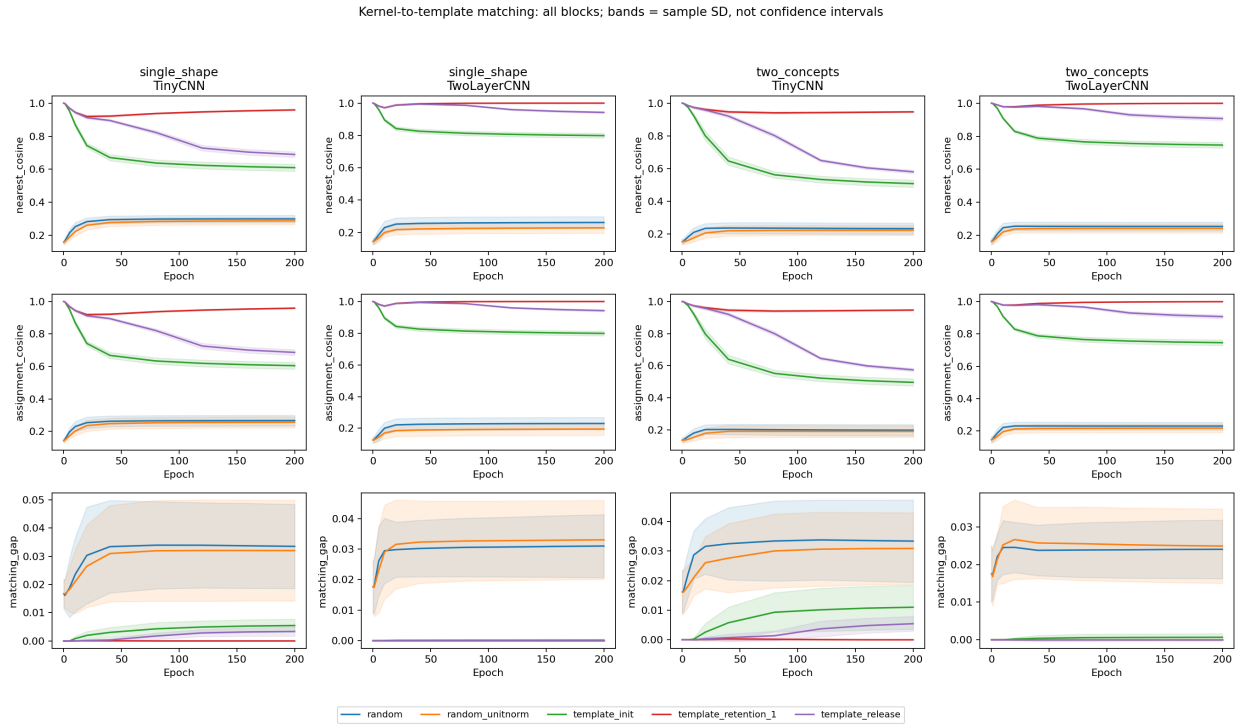

Figure 7: Nearest-template alignment, maximum-weight one-to-one alignment, and their gap over all saved epochs. All five conditions and four settings are shown. Bands are sample SD over ten blocks, not confidence intervals.

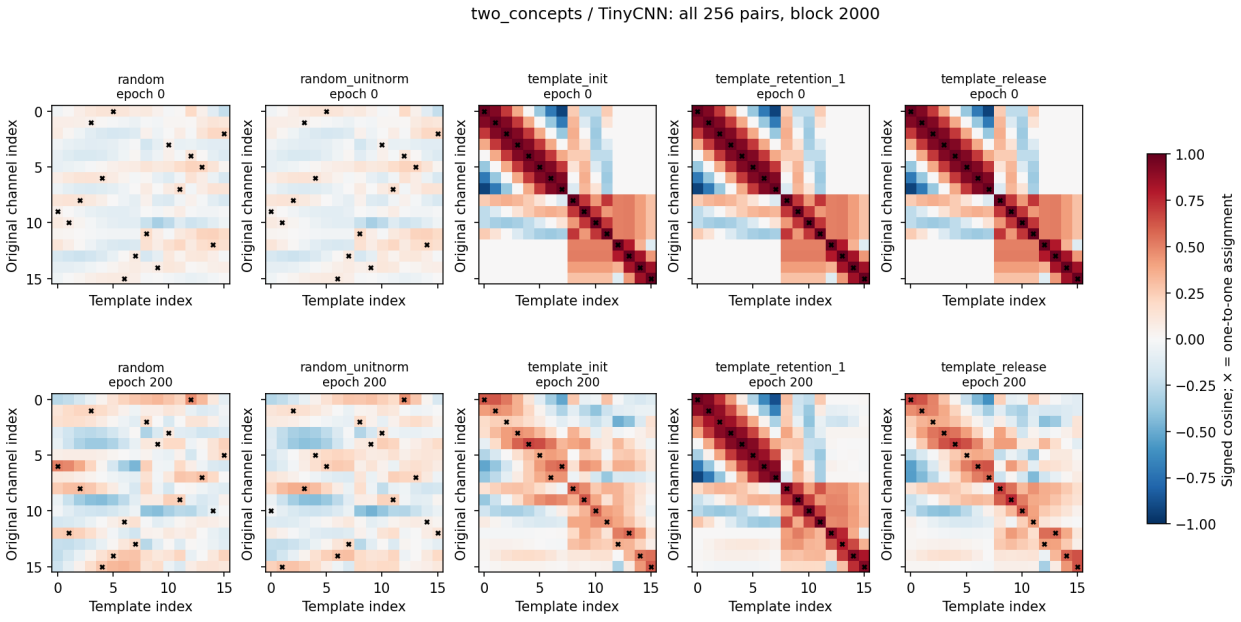

Figure 8: All 256 signed kernel/template similarities for each condition at initialization and epoch 200, two_concepts/TinyCNN, fixed block 2000. Rows retain original channel indices, columns retain template indices; black crosses mark the optimal one-to-one assignment. Values range from −1 to 1. Matrices and galleries for all settings are retained in the derived analysis artifacts.