RoSeViT: Role-Separated Vision Transformers for ARC Visual Reasoning

Zhiguang Liu
Department of Electrical Engineering and Computer Science
University of Missouri
Columbia, MO 65211, USA
lz7fd@missouri.edu

Yi Shang
Department of Electrical Engineering and Computer Science
University of Missouri
Columbia, MO 65211, USA
shangy@missouri.edu

Abstract

ARC tasks can be formulated as visual transformation problems, where a model must infer a high-level transformation rule from a small set of demonstrations and apply it consistently across a grid. While dense Vision Transformers (ViTs) perform well in this setting, they entangle local visual state and global transformation logic within a shared token representation, which can hinder consistent reasoning. In this paper, we introduce RoSeViT, a role-separated Vision Transformer that explicitly decouples high-level instruction inference from visual workspace computation. RoSeViT uses controller tokens to aggregate transformation-level information and workspace tokens to represent the visual grid, with a structured attention pass in which nonlocal workspace interactions are routed through the controller. This design is motivated by our formal analysis showing that enforcing a shared instruction representation can reduce the effective complexity of visual transformation modeling. RoSeViT outperforms dense ViT baselines under the standard VARC training and evaluation protocol for ARC. In a controlled backbone replacement setting, it improves ARC-1 accuracy from 54.5 to 56.6, and with recurrent refinement reaches 58.8. When integrated into the full system, RoSeViT-Ensemble improves performance from 60.4 to 62.6 on ARC-1. These results demonstrate that explicitly separating instruction and workspace representations leads to more effective visual reasoning on ARC tasks. GitHub repository: https://github.com/lz7fd/RoSeViT.

1 Introduction

ARC tasks ask a system to infer a transformation from a few input-output demonstrations and apply it to a query grid [Chollet, 2019]. Many tasks can be viewed as visual transformations governed by a high-level instruction: copy an object, mirror a shape, extend a pattern, or recolor matching cells. A successful solver must infer this instruction from demonstrations and use it consistently across the visual workspace.

VARC shows that ARC can be approached as vision-conditioned grid-to-grid prediction with test-time adaptation and multi-view voting [Hu et al., 2026]. Dense Vision Transformers [Dosovitskiy et al., 2021] are effective in this setting. However, they assign the same computational state to all patch tokens: each patch token may carry local grid state and high-level transformation instruction simultaneously. This gives dense ViT a larger instruction-state space than visual transformations require.

Biological and cognitive evidence for separated slow/global and fast/local processing [Murray et al., 2014, Zeraati et al., 2023, Huntenburg et al., 2018, Başar et al., 2001, Buzsáki, 2006, Pahor and

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Jaušovec, 2014, Tort et al., 2009], together with recent two-channel reasoning systems such as HRM and TRM [Wang et al., 2025, Jolicoeur-Martineau, 2025], suggests that the single-state dense ViT may be suboptimal for visual transformation problems. We formalize this with the Controller-Route Entropy Reduction Theorem: adding a controller route decreases the entropy of the available shared-rule solution route while preserving dense ViT expressivity. The theorem motivates architectures that keep dense visual feature formation and add an enforced high-level instruction route.

Motivated by this theorem, we introduce RoSeViT, a role-separated Vision Transformer for ARC visual reasoning. RoSeViT uses controller tokens to infer high-level transformation evidence from demonstrations and workspace tokens to maintain the visual grid state. Each block combines dense visual feature formation with a structured controller-mediated pass, so nonlocal workspace communication factors through the controller. Proposition 1 shows that this route is globally low-rank.

Under the same VARC training and testing protocol, replacing the dense VARC-ViT backbone with one-step RoSeViT improves ARC-1 accuracy from 54.5 to 56.6. Architecture ablations show that the controller bottleneck must be paired with dense visual feature formation, attention maps are evidence for controller-mediated routing, and recurrent refinement improves the best single RoSeViT model to 58.8$\pm$0.7 on ARC-1 and 11.0$\pm$1.0 on ARC-2. RoSeViT-Ensemble reaches 62.6 on ARC-1 and 13.5 on ARC-2, improving VARC-Ensemble from 60.4 to 62.6 on ARC-1.

Our contributions are:

- **Controller-Route Entropy Reduction Theorem.** We show that adding a controller route reduces the available solution-route entropy for shared-rule visual transformations from $\Theta(Ld\log(1/\epsilon))$ to $\Theta(d\log(1/\epsilon))$, while RoSeViT contains the dense ViT route; whether a trained model uses the route is tested empirically.
- **RoSeViT.** Motivated by this theorem, we introduce a role-separated Vision Transformer that realizes the controller route through controller instruction tokens and visual workspace tokens. Under the same VARC protocol, RoSeViT improves dense VARC-ViT from 54.5 to 56.6 in the controlled backbone comparison, reaches 58.8 as the best single model, and improves VARC-Ensemble from 60.4 to 62.6 on ARC-1.

2 Related Work

ARC and visual reasoning. ARC was introduced to measure skill-acquisition and abstraction from a small number of demonstrations [Chollet, 2019], with ARC-AGI-2 continuing this line as a benchmark for frontier reasoning systems [Chollet et al., 2025]. Human studies show that ARC remains a demanding visual reasoning benchmark even for people [LeGris et al., 2024]. Recent progress comes from treating ARC as a vision-conditioned grid-to-grid problem: RE-ARC expands task demonstrations procedurally [Hodel, 2024], and VARC combines visual augmentation, test-time adaptation, and multi-view voting [Hu et al., 2026]. RoSeViT keeps this evaluation stack and changes the backbone, making the controlled comparison in Section 5 an architectural test.

Dense ViTs and token roles. Transformers [Vaswani et al., 2017] and Vision Transformers [Dosovitskiy et al., 2021] use dense self-attention over tokens, and VARC shows that this design is already strong for ARC visual reasoning [Hu et al., 2026]. Dense ViTs can model grids, yet their patch tokens share the same computational state: every patch token is free to carry both the inferred transformation and the local workspace state. Section 3 formalizes why this creates a larger instruction-state class than visual transformations require.

Bottlenecks and special tokens. Many architectures use special or latent tokens to aggregate information, including ViT class tokens [Dosovitskiy et al., 2021], Perceiver-style latent arrays [Jaegle et al., 2021, 2022], Set Transformer inducing points [Lee et al., 2019], Slot Attention slots [Locatello et al., 2020], and ViT register tokens [Darcet et al., 2024]. These works establish that token roles and bottlenecks can be useful computational primitives. RoSeViT uses that design space for a different ARC-specific purpose: the controller bottleneck is required to infer one high-level transformation instruction and make it available to the workspace through a structured pass in which nonlocal workspace communication goes through the controller.

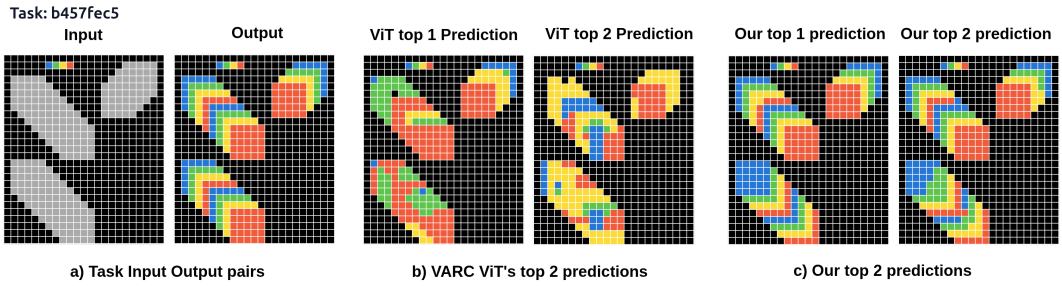


Figure 1: ARC visual reasoning is transformation-rule induction, more than visual completion. VARC-ViT produces locally plausible colored structures, while RoSeViT c-d-4-TTT3 more consistently applies the inferred transform across repeated objects.

Hierarchical and recurrent reasoning. Biological evidence suggests separated slow/global and fast/local processing timescales [Murray et al., 2014, Zeraati et al., 2023, Huntenburg et al., 2018, Başar et al., 2001, Buzsáki, 2006, Pahor and Jaušovec, 2014, Tort et al., 2009]. HRM and TRM bring related hierarchical or recurrent ideas to ARC-style reasoning [Wang et al., 2025, Jolicoeur-Martineau, 2025]. Our contribution is the computational account missing from these motivations: the Controller-Route Entropy Reduction Theorem (Theorem 1) shows that adding a controller route decreases solution-route entropy while preserving the dense route, and the collapse condition explains why nominal two-channel systems still need an enforced bottleneck. Recurrence in RoSeViT is therefore a performance amplifier applied after the role-separated backbone is defined.

Test-time adaptation. Test-time training improves adaptation under limited supervision and distribution shift [Sun et al., 2020, Akyürek et al., 2025]. VARC uses this idea for ARC task adaptation [Hu et al., 2026], and RoSeViT follows the same protocol. The central attribution claim is therefore the same-protocol backbone comparison in Table 1; stronger TTT settings and ensembling are reported as part of the full ARC evaluation stack.

3 Visual Transformation Complexity

3.1 Entangled and role-separated routes

ARC is transformation-rule induction, more than visual completion. Figure 1 shows a dense ViT producing locally plausible colored objects while failing to apply the same transform consistently. In a dense ViT, each output patch is decoded from a patch token in the same computational stream, so local visual state and transformation-rule information are represented together. RoSeViT keeps this dense visual stream and adds controller tokens as a separated route for transformation-level information.

This design is conservative. The dense route preserves the original VARC-ViT computation, while the controller route gives shared transformation information a lower-complexity path through the model. We measure this by solution-route entropy: the metric entropy of an architecture-provided route that can realize the shared-rule transformation.

3.2 Controller-route entropy reduction

Theorem 1 (Controller-Route Entropy Reduction Theorem)**.** *Let visual transformations be governed by one d-dimensional shared rule over $L > 1$ workspace sites. Assume the rule-to-output maps are bi-Lipschitz in the rule variable. A dense single-state patch route represents the rule through site-wise states*

$$\mathcal{F}_{\text{site}} = \{F_{z_1,\dots,z_L} : z_i \in [0,1]^d\},$$

with metric entropy

$$\log N(\epsilon, \mathcal{F}_{\text{site}}, d_\infty) = \Theta\left(Ld \log \frac{1}{\epsilon}\right).$$

RoSeViT preserves this dense route and adds a controller route

$$\mathcal{F}_{\mathrm{ctrl}} = \{F_c : c \in [0,1]^d\},$$

with metric entropy

$$\log N(\epsilon, \mathcal{F}_{\mathrm{ctrl}}, d_\infty) = \Theta\left(d \log \frac{1}{\epsilon}\right).$$

For a set of available solution routes $\mathfrak{R}$, define

$$\mathcal{H}_\epsilon(\mathfrak{R}) = \min_{\mathcal{F} \in \mathfrak{R}} \log N(\epsilon, \mathcal{F}, d_\infty).$$

Then the dense ViT route set $\mathfrak{R}_{\mathrm{dense}} = \{\mathcal{F}_{\mathrm{site}}\}$ satisfies

$$\mathcal{H}_\epsilon(\mathfrak{R}_{\mathrm{dense}}) = \Theta\left(Ld \log \frac{1}{\epsilon}\right),$$

whereas the RoSeViT route set $\mathfrak{R}_{\mathrm{rose}} = \{\mathcal{F}_{\mathrm{site}}, \mathcal{F}_{\mathrm{ctrl}}\}$ satisfies

$$\mathcal{H}_\epsilon(\mathfrak{R}_{\mathrm{rose}}) = \Theta\left(d \log \frac{1}{\epsilon}\right),$$

and the full RoSeViT class contains the dense ViT class:

$$\mathcal{F}_{\mathrm{dense}} \subseteq \mathcal{F}_{\mathrm{rose}}.$$

Here $\mathcal{F}_{\mathrm{dense}}$ and $\mathcal{F}_{\mathrm{rose}}$ denote the full dense-ViT and RoSeViT function classes. Thus adding the controller route decreases the available solution-route entropy by the workspace-size factor L while preserving dense ViT expressivity.

The theorem also explains why a second channel alone is insufficient. The entropy reduction is obtained only when the architecture enforces the controller route; if workspace tokens can freely carry independent site-wise rule states, the effective solution route can collapse back to $\mathcal{F}_{\mathrm{site}}$. Appendix A formalizes this collapse condition.

We use the theorem solely as motivation: it states that a lower-entropy route exists once the controller route is added, not that optimization selects it, that the controller encodes the intended rule, or that reduced entropy improves generalization. Those are empirical claims, tested in Section 5.4. Section 4 gives the controller-mediated mechanism, Section 5 tests concrete instantiations, and Appendix A provides the proof.

4 RoSeViT

RoSeViT is designed to instantiate the controller-route entropy reduction in Theorem 1. The architecture must satisfy three requirements: preserve the dense visual route, infer a high-level instruction from demonstrations, and enforce a bottleneck that gives this instruction a separated route to the workspace. We keep the VARC visual formulation, augmentation protocol, and output projection, but replace the dense ViT backbone with a role-separated Transformer block. Figure 2 summarizes the full pipeline.

4.1 Visual workspace

An ARC task T provides a small demonstration set $D^T_{\mathrm{demo}} = \{(x_i, y_i)\}_{i=1}^m$ and a query input x_{infer}. Following VARC, each grid is embedded into a larger augmented canvas using task-preserving transformations such as rotation, translation, scaling, and fixed borders. Let $\tilde{x}$ denote the augmented query canvas. We tokenize $\tilde{x}$ using non-overlapping $p \times p$ patches and map each patch to a D-dimensional vector with a linear patch embedding and a 2D positional embedding. This yields workspace tokens

$$w = [w_1, \ldots, w_L] \in \mathbb{R}^{L \times D}.$$

In our experiments, the canvas is 64×64 and $p = 2$, so $L = 1024$.

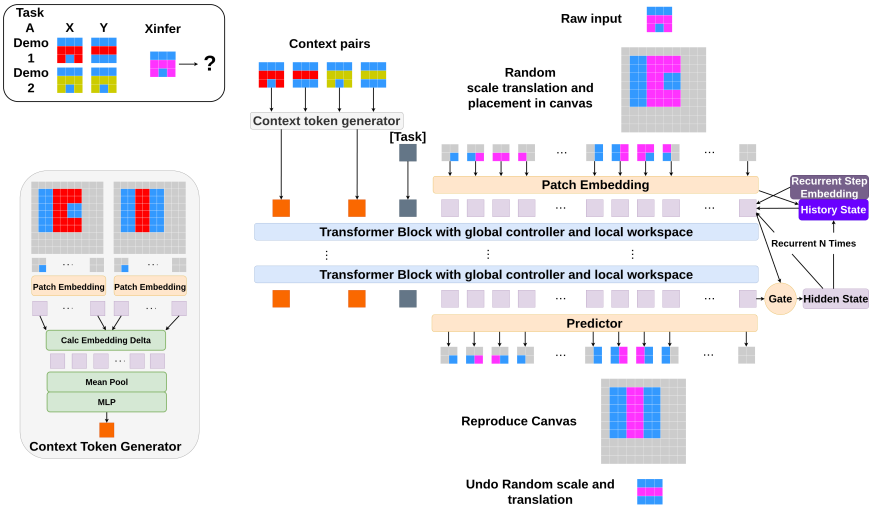


Figure 2: Overview of RoSeViT in the VARC visual reasoning pipeline. Demonstration input-output pairs are transformed into compact controller tokens, while the query grid is tokenized as a local workspace. RoSeViT blocks repeatedly combine dense visual feature formation with a structured controller-mediated pass, and the predicted canvas is mapped back to the ARC output grid by inverse augmentation.

4.2 Controller instruction tokens

The global controller consists of one task token and one context token per demonstration pair. The task token τ_T is learned during offline training for training tasks; for a new test task, a fresh task token is adapted during test-time training.

For each demonstration pair (x_i, y_i), we apply the same augmentation family used for the query grid and embed the augmented input and output grids into patch tokens $E(\tilde{x}_i), E(\tilde{y}_i) \in \mathbb{R}^{L\times D}$. We form a pooled transformation descriptor and project it to a context token:

$$\bar{\Delta}_i = \mathrm{Pool}(E(\tilde{y}_i) - E(\tilde{x}_i)), \qquad c_i = \mathrm{MLP}_{\mathrm{ctx}}(\bar{\Delta}_i).$$

The controller sequence is

$$g = [\tau_T, c_1, \ldots, c_m] \in \mathbb{R}^{P\times D}, \qquad P = m + 1,$$

and the initial token sequence is

$$h^{(0)} = [g; w] \in \mathbb{R}^{(P+L)\times D}.$$

The controller tokens provide the high-level instruction route: they aggregate demonstration-level transformation evidence and make it available to the query workspace. Workspace tokens keep the visual state that will be locally updated under this instruction.

4.3 Enforced controller bottleneck

Each RoSeViT block contains a dense visual pass, a structured role-separated pass, and an MLP. The dense pass preserves ViT feature formation. The structured pass then re-imposes role separation: controller queries attend globally, while workspace queries attend only to the controller tokens and to a small local workspace neighborhood that includes the token itself,

$$\mathcal{A}_{\mathrm{ws}}(i) = \{1, \ldots, P\} \cup \{P + j : j \in \mathcal{N}_k(i) \cup \{i\}\},$$

where $\mathcal{N}_k(i)$ is the 2D local neighborhood of workspace site i. In our main model, $k = 8$, corresponding to the eight neighboring patches. Thus, in the structured pass, high-level instruction information reaches every workspace token and nonlocal workspace-to-workspace communication passes through the compact controller. This is the architectural constraint corresponding to the collapse condition in Section 3: the attention topology of the structured pass re-imposes role separation through the controller bottleneck.

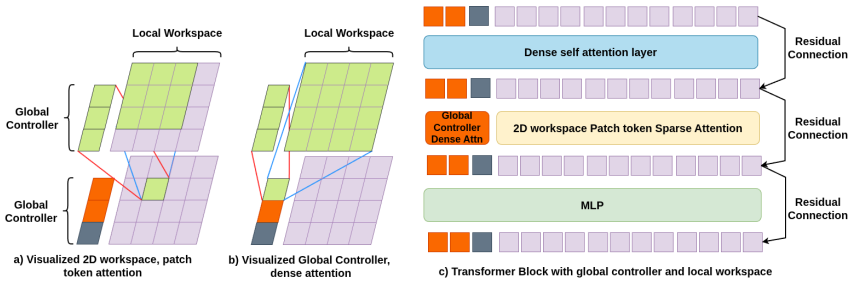


Figure 3: Role-separated Transformer block. Controller tokens attend globally, while workspace tokens attend to the controller and a local spatial neighborhood. The structured pass repeatedly reimposes a controller-mediated bottleneck after dense visual feature mixing.

The structured pass gives the added route in Theorem 1: nonlocal workspace influence factors through the controller, giving the high-level instruction a designated path through the model. Formally, this route is globally low-rank and uses $E_{\text{role}} = P(P+L) + L(P+k+1)$ structured-pass edges, compared with $E_{\text{dense}} = (P+L)^2$ dense edges; the full statement and proof are in Appendix B. With $L = 1024$, $P \leq 5$, and $k = 8$, each workspace token attends to at most $P + k + 1 \leq 14$ tokens in the structured pass instead of $P + L = 1029$, and the block uses about 1.84% as many structured-pass edges as dense attention. This edge count describes the route, not our implementation, which runs the structured pass as dense attention under a mask on top of the dense pass (measured cost in Section 5).

4.4 Recurrent refinement

We optionally wrap the RoSeViT encoder in a lightweight recurrent refinement loop to increase effective depth without increasing parameter count. Let f_θ denote the stack of role-separated Transformer blocks. At recurrent step r, the model maintains a current token state $h^{(r)}$ and a history state $s^{(r)}$. It computes

$$u^{(r)} = f_\theta(h^{(r)} + s^{(r)}), \qquad \gamma^{(r)} = \sigma\Big(W_g[h^{(r)}; s^{(r)}]\Big),$$

and applies a gated residual update

$$h^{(r+1)} = h^{(r)} + \gamma^{(r)} \odot u^{(r)}.$$

The history state is updated by an exponential moving average,

$$s^{(r+1)} = \alpha s^{(r)} + (1-\alpha) h^{(r+1)}.$$

This recurrence implements iterative refinement over a fixed ARC query rather than temporal sequence modeling.

4.5 Prediction and inverse augmentation

After the final block or recurrent step, RoSeViT discards the controller tokens for output decoding and applies a linear head to each workspace token. The head predicts categorical logits for the corresponding $p \times p$ patch on the augmented canvas. We assemble the patch logits into a full canvas prediction, invert the augmentation, crop to the original grid support, and decode the final ARC output by color-wise argmax.

5 Experiments

The experiments test the architectural prediction motivated by the Controller-Route Entropy Reduction Theorem: adding a separated high-level instruction route while preserving dense visual computation should improve visual transformation learning. We follow the VARC visual ARC protocol throughout. Results are pass@2 accuracy on ARC-1 and ARC-2. RoSeViT results are reported as mean over 4 seeds. Offline training uses ARC-1 training tasks with RE-ARC expansion; ARC-2 is not used for offline pretraining. At test time, each task is adapted using its demonstrations and evaluated with 510-view voting. Full protocol, variant, TTT, and inference details are in Appendix C.

Table 1: Same-training/same-testing backbone comparison (VARC default adaptation, ARC-1 pass@2) with a parameter-matched dense control and the measured forward cost (batch 8, H100). Theoretical GFLOPs count the sparse structured pass; actual GFLOPs are those of our masked dense implementation.

Model	#param	ARC-1	GFLOPs theor. / actual	Latency
VARC-ViT (dense)	18M	54.5	53.8 / 53.8	20.7 ms
VARC-ViT, width-matched (dense)	28M	53.6	79.4 / 79.4	22.2 ms
RoSeViT d-1 (role-separated)	28M	**56.6**	75.9 / 97.3	46.1 ms

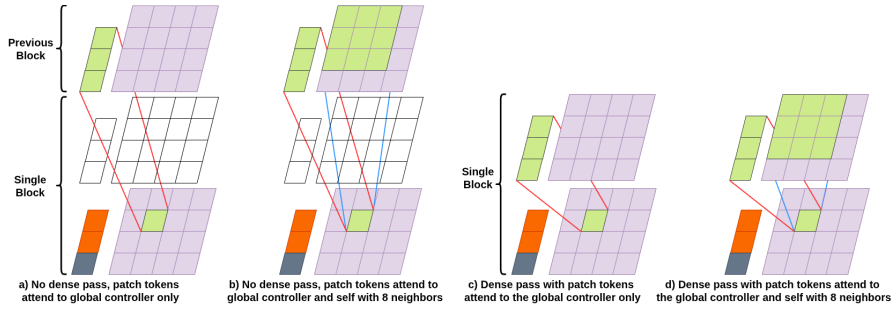


Figure 4: RoSeViT block variants used in the architecture ablation. Variants a/b use only the structured pass, while variants c/d first perform dense visual feature mixing and then re-impose the controller-mediated bottleneck.

5.1 Same-protocol test of role separation

Theorem 1 motivates the controlled test: replace the dense VARC-ViT backbone with a one-step RoSeViT d-1 block while keeping the same training and testing protocol. The input representation, offline data, augmentation, adaptation budget, 510-view inference, voting rule, and pass@2 metric are unchanged.

Under this controlled setting, RoSeViT d-1 improves ARC-1 accuracy from 54.5 to 56.6, a +2.1 point gain over VARC-ViT. This is the central empirical test of the high-level-instruction design: with training and testing fixed, the gain targets the effect of replacing entangled patch-token transformation representation with controller-mediated instruction routing and workspace execution.

Parameter-matched control. RoSeViT d-1 has 28M parameters against 18M for VARC-ViT. To separate role separation from capacity, Table 1 also reports a dense VARC-ViT widened to 28M under the identical protocol (Appendix D.1). Widening does not help: the control reaches 53.6 ± 1.0 over 8 adaptation seeds (range 52.1–54.9), in line with the flat-to-declining dense trend $54.5 \rightarrow 53.6 \rightarrow 53.0$ at 18M, 28M, and 66M, and its best seed (54.9) is below the RoSeViT mean. The +2.1 gain over VARC-ViT and the +3.0 gain over the matched control are therefore architectural.

Compute and latency. The structured pass is added on top of the dense pass, so RoSeViT is not a cheaper model. Counted at its sparse edge set (Section 4), its theoretical forward cost is 75.9 GFLOPs, below the 79.4 GFLOPs of the matched dense control; but no kernel for this access pattern was developed in this work, our implementation runs the structured pass as dense attention under a mask, and the actual cost is 97.3 GFLOPs at about $2.1\times$ the latency of the matched control (Table 1, Appendix D.4). The claim of this paper is role separation, not speed.

5.2 Which architecture realizes the instruction route?

The collapse condition says that high/low channels are useful only when the architecture enforces the instruction route. Figure 4 therefore tests four ways to instantiate the theorem: a/b use only the structured controller/workspace pass, while c/d first perform dense visual feature formation and then re-impose the controller-mediated bottleneck.

The ablation shows why the architecture must realize both sides of the theory. Pure bottleneck variants a/b enforce separation but are too restrictive for ARC visual modeling. Variants c/d restore dense visual feature formation and then re-impose the bottleneck; d adds local workspace neigh-

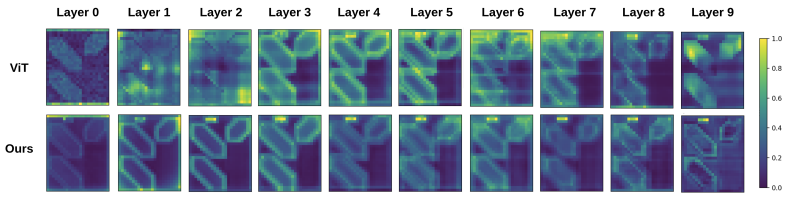


Figure 5: Layer-wise attention maps comparing a dense VARC-ViT backbone and RoSeViT d-1-TTT1 under the same training and testing setting. RoSeViT d-1-TTT1 preserves sharper spatial primitives across depth, matching controller-mediated high-level instruction routing and local workspace execution.

borhoods and gives the strongest one-step backbone. Training curves supporting this conclusion are reported in Appendix C.3. The effective instantiation is dense visual modeling followed by a structured controller/workspace route.

5.3 Visible signature of controller-mediated routing

Figure 5 is evidence that the implemented RoSeViT block forms the controller-mediated route characterized by Proposition 1. Dense ViT attention diffuses through the visual workspace, consistent with patch tokens mixing local state and transformation instruction. RoSeViT forms controller-mediated routing: high-level instruction information is routed through controller tokens, while workspace tokens preserve sharper spatial primitives for local grid updates.

5.4 Controller-route evidence: ablation and probe

Whether the trained model uses the controller route is an empirical question; Table 2 reports two tests on RoSeViT d-1 and the matched dense control (protocols in Appendix D.2). First, an inference-time ablation zeroes the controller embeddings, leaving weights, masks, dense pass, and local neighborhoods untouched. Removing all conditioning collapses both models, so collapse itself is not route-specific; the discriminating results are rule-specificity and non-redundancy. Given the controllers of a different task, RoSeViT follows the controllers rather than the input (2.9% versus 81.3%), and the task token and the demonstration-derived context tokens are individually near-essential. After test-time adaptation, with intact and ablated predictions from the same adapted weights, the dense control barely needs its conditioning (-2.5 pass@2), whereas RoSeViT still routes rule application through its controllers (-13.1). Second, a linear probe (task token excluded) finds the demonstration-defined rule about twice as decodable from the context tokens (64.1/53.8/44.0/30.1% at layers 1/3/6/10) as from the workspace tokens (33.3/27.7/24.2/16.8%; chance 0.25%).

These results establish the controller route as realized, causally necessary, and rule-specific. They do not establish that the bottleneck prevents rule information from appearing in workspace tokens: the probe decodes the rule from workspace tokens well above chance. The measured form of the enforcement claim is that the controller route is necessary and workspace computation cannot replace it, not that the workspace carries no rule state.

5.5 Recurrence and full ARC performance

After establishing the one-step role-separated backbone, we add recurrence as a standard performance amplifier for ARC-style reasoning systems, aligned with the practice in HRM and TRM. Recurrence increases effective computation without increasing parameter count. The c-2 model applies two recurrent refinement steps; c-d-4 uses staged training, first forming the controller-only route and then enabling local workspace neighborhoods.

Table 3a summarizes the main trained variants. The d-1 rows keep the one-step architecture fixed and stay within a narrow ARC-1 range from 55.5 to 56.6 across adaptation settings. The larger gains appear when controller-mediated instruction routing is refined recurrently: c-2-TTT2 reaches 58.0 on ARC-1, and c-d-4-TTT3 reaches 58.8$\pm$0.7 on ARC-1 and 11.0$\pm$1.0 on ARC-2.

ARC-2 under the controlled setting. Under the VARC default adaptation (TTT1), d-1 scores 6.9 on ARC-2 against 8.3 for VARC-ViT. Both backbones are ARC-1 priors (offline training uses ARC-1

Table 2: Controller-route evidence. Top: inference-time ablation, exact match % on held-out pairs of training tasks (the dense control has no context tokens). Middle: ARC-1 pass@2 after adaptation, intact versus ablated predictions from the same weights (two seeds). Bottom: linear rule probe, decodability % of the demonstration-defined rule from context versus workspace tokens of RoSeViT d-1 by layer (chance 0.25%).

Inference condition (offline, exact match %)	RoSeViT d-1	Dense 28M
intact	81.3	77.9
all conditioning zeroed (task + context / task)	0.0	0.0
wrong-task controllers	2.9	–
task token only zeroed	4.3	0.0
context tokens only zeroed	5.5	–
After TTT (ARC-1 pass@2, 2 seeds)	**intact → ablated**	**Δ**
RoSeViT d-1 (controllers zeroed)	56.3 / 56.3 → 43.3 / 43.0	−13.1
Dense 28M (task token zeroed)	52.9 / 54.9 → 50.3 / 52.5	−2.5
Rule probe, RoSeViT d-1 (%; chance 0.25)	**context tokens**	**workspace tokens**
layers 1 / 3 / 6 / 10	64.1 / 53.8 / 44.0 / 30.1	33.3 / 27.7 / 24.2 / 16.8

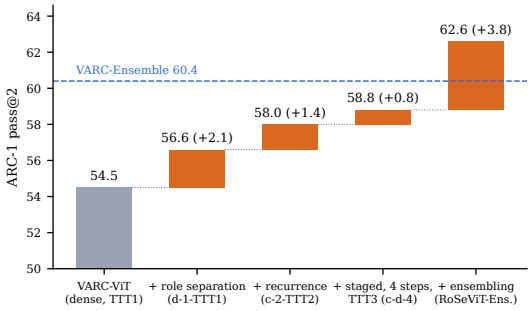


Figure 6: ARC-1 gain along the variant ladder of Table 3; each step is the difference between consecutive rows, so the recurrence and staged steps also carry their adaptation-setting change (TTT1 → TTT2 alone moves d-1 by −0.2). Dashed line: VARC-Ensemble.

tasks with RE-ARC expansion; ARC-2 is never seen): where the prior applies, role separation extracts more from it, while under the ARC-2 distribution shift neither prior confers an advantage and performance is set by the adaptation budget. Accordingly, d-1's ARC-2 score rises with adaptation strength, 6.9 → 7.7 → 8.3 from TTT1 to TTT3, reaching parity before any recurrence, and the full system reverses the gap (11.0 versus 8.3). No single adaptation setting improves both benchmarks for d-1: the setting that reaches ARC-2 parity (TTT3) costs 1.1 points on ARC-1.

Gain decomposition. Figure 6 decomposes the ARC-1 improvement along the ladder of Table 3: role separation alone (the controlled swap) +2.1, recurrence +1.4, staged training with four recurrent steps and TTT3 +0.8, ensembling +3.8. Voting is part of the fixed protocol in every row, each amplifier is shared machinery available to both backbones, and the like-for-like ensemble comparison is 60.4 versus 62.6. The controlled +2.1 remains the central claim.

5.6 Comparison with prior methods

Table 3b compares RoSeViT with recent ARC solvers and connects the empirical findings back to the theory. RoSeViT improves over the dense VARC-ViT backbone, while the larger VARC-ViT underperforms the default dense model. This supports the thesis that scaling the dense route alone does not add the separated shared-rule route. HRM and TRM use hierarchical or recurrent structure; their lower scores are consistent with, but not evidence for, the collapse condition, since those systems differ from RoSeViT in many respects beyond whether a controller-mediated bottleneck is enforced.

The best single RoSeViT model reaches 58.8±0.7 on ARC-1 and 11.0±1.0 on ARC-2, improving over VARC-ViT by +4.3 and +2.7 points. RoSeViT-Ensemble reaches 62.6 on ARC-1 and 13.5 on ARC-2, improving VARC-Ensemble from 60.4 to 62.6 on ARC-1. The ensemble result is also competitive with the reported average human score on ARC-1 [LeGris et al., 2024]; the central scientific

Table 3: RoSeViT variants and comparison with prior ARC methods (pass@2). All RoSeViT models use the VARC visual protocol; the controlled architecture comparison is d-1-TTT1 versus VARC-ViT, and c-d-4-TTT3 is the strongest single RoSeViT model.

(a) RoSeViT variants under different recurrent depths and TTT strategies; the rows form the ladder plotted in Figure 6.

Model	#param	ARC-1	ARC-2
VARC-ViT	18M	54.5	8.3
VARC-ViT (matched)	28M	53.6	–
VARC-ViT-large	66M	53.0	–
d-1-TTT1	28M	56.6	6.9
d-1-TTT2	28M	56.4	7.7
d-1-TTT3	28M	55.5	8.3
c-2-TTT2	28M	58.0	8.3
c-d-4-TTT3	28M	**58.8**	**11.0**

(b) Comparison to prior ARC methods. RoSeViT improves the single-model VARC-ViT backbone and remains compatible with VARC-Ensemble.

Method	#param	ARC-1	ARC-2
HRM [Wang et al., 2025]	27M	40.3	5.0
TRM [Jolicoeur-Martineau, 2025]	7M	44.6	7.8
VARC-ViT [Hu et al., 2026]	18M	54.5	8.3
VARC-ViT (matched, ours)	28M	53.6	–
VARC-ViT-large [Hu et al., 2026]	66M	53.0	–
VARC-Ensemble [Hu et al., 2026]	73M	60.4	11.1
RoSeViT c-d-4-TTT3	28M	**58.8**	11.0
RoSeViT-Ensemble	83M	**62.6**	**13.5**

evidence remains the same-protocol comparison, the architecture ablation, and the controller-route evidence above.

6 Discussion and Limitations

This paper establishes a specific claim about visual transformation systems: role separation can preserve dense visual computation while adding a lower-complexity route for shared transformation information. The Controller-Route Entropy Reduction Theorem motivates adding this route, while RoSeViT gives a concrete ViT realization through controller instruction tokens and visual workspace tokens. The central empirical evidence is the same-protocol backbone comparison, where RoSeViT d-1 improves over dense VARC-ViT with the VARC training, adaptation, voting, and evaluation pipeline held fixed; the full-system results of Table 3b build on it.

The architecture ablation clarifies what kind of role separation is useful. Pure controller bottlenecks enforce separation but remove too much visual capacity for ARC grids. The effective design combines dense visual feature formation with a structured controller-mediated pass that re-imposes the instruction route. This also sets the claim boundary: recurrence and ensembling improve the full ARC system, while the central evidence is the effectiveness of the role-separated Vision Transformer under the same protocol.

The claims of this paper are confined to ARC-type visual transformation reasoning with a shared per-task rule. The theorem assumes one shared d-dimensional rule; conditional, object-specific, or compositional transformations exceed this assumption and are untested, as is whether reduced route entropy correlates quantitatively with generalization. Transfer beyond ARC is preliminary: an instruction-conditioned maze suite (Appendix E) shows the same qualitative advantage under a feedforward, offline protocol, but it is a new suite with two seeds and several rules unsolved by either model, and we make no claims about embodied or real-world visual tasks. RoSeViT is not a cheaper model: without a sparse kernel our implementation runs at about $2.1\times$ the latency of the matched dense control. RoSeViT also inherits VARC's test-time training and multi-view voting stack; the controlled comparison isolates the backbone, while final performance depends on this pipeline. Finally, the controller instruction is latent: the ablation and probe show the route is necessary and rule-specific, but rule information remains decodable from workspace tokens, so the bottleneck shapes rather than fully prevents site-wise rule state.

Acknowledgements and Disclosure of Funding

This work received no external funding. Zhiguang Liu was supported by a graduate research assistantship from the University of Missouri, Columbia; the compute used for the experiments was paid for by the authors. The authors declare no competing interests.

References

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pages 942–963, 2025.

Erol Başar, Canan Başar-Eroglu, Sirel Karakaş, and Martin Schürmann. Gamma, alpha, delta, and theta oscillations govern cognitive processes. *International Journal of Psychophysiology*, 39(2-3): 241–248, 2001.

György Buzsáki. *Rhythms of the Brain*. Oxford University Press, 2006.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Francois Chollet, Mike Knoop, Gregory Kamradt, Bryan Landers, and Henry Pinkard. ARC-AGI-2: A new challenge for frontier AI reasoning systems. *arXiv preprint arXiv:2505.11831*, 2025.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations*, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Michael Hodel. Addressing the abstraction and reasoning corpus via procedural example generation, 2024. URL `https://arxiv.org/abs/2404.07353`.

Keya Hu, Ali Cy, Linlu Qiu, Xiaoman Delores Ding, Runqian Wang, Yeyin Eva Zhu, Jacob Andreas, and Kaiming He. ARC is a vision problem! In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Julia M Huntenburg, Pierre-Louis Bazin, and Daniel S Margulies. Large-scale gradients in human cortical organization. *Trends in Cognitive Sciences*, 22(1):21–31, 2018.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, pages 4651–4664. PMLR, 2021.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Henaff, Matthew Botvinick, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver IO: A general architecture for structured inputs and outputs. In *International Conference on Learning Representations*, 2022.

Alexia Jolicoeur-Martineau. Less is more: Recursive reasoning with tiny networks, 2025. URL `https://arxiv.org/abs/2510.04871`.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-Supervised Nets. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 562–570, San Diego, California, USA, 09–12 May 2015. PMLR. URL `https://proceedings.mlr.press/v38/lee15a.html`.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pages 3744–3753. PMLR, 2019.

Solim LeGris, Wai Keen Vong, Brenden M Lake, and Todd M Gureckis. H-ARC: A robust estimate of human performance on the abstraction and reasoning corpus benchmark. *arXiv preprint arXiv:2409.01374*, 2024.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, pages 11525–11538, 2020.

John D Murray, Alberto Bernacchia, David J Freedman, Ranulfo Romo, Jonathan D Wallis, Xinying Cai, Camillo Padoa-Schioppa, Tatiana Pasternak, Hyojung Seo, Daeyeol Lee, et al. A hierarchy of intrinsic timescales across primate cortex. *Nature Neuroscience*, 17(12):1661–1663, 2014.

Anja Pahor and Norbert Jaušovec. Theta–gamma cross-frequency coupling relates to the level of human intelligence. *Intelligence*, 46:283–290, 2014.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pages 9229–9248. PMLR, 2020.

Adriano BL Tort, Robert W Komorowski, Joseph R Manns, Nancy J Kopell, and Howard Eichenbaum. Theta–gamma coupling increases during the learning of item–context associations. *Proceedings of the National Academy of Sciences*, 106(49):20942–20947, 2009.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. Hierarchical reasoning model, 2025. URL `https://arxiv.org/abs/2506.21734`.

Roxana Zeraati, Yan-Liang Shi, Nicholas A Steinmetz, Marc A Gieselmann, Alexander Thiele, Tirin Moore, Anna Levina, and Tatiana A Engel. Intrinsic timescales in the visual cortex change with selective attention and reflect spatial connectivity. *Nature Communications*, 14(1):1858, 2023.

A Proof and Collapse Condition for the Controller-Route Entropy Reduction Theorem

We prove the theorem in three steps. First, RoSeViT contains the dense route. Let a dense ViT block be written as D_θ and the corresponding RoSeViT block as

$$R_{\theta,\psi}(h) = D_\theta(h) + S_\psi(h),$$

where S_ψ is the residual controller-mediated route. For any dense ViT function $f \in \mathcal{F}_{\text{dense}}$, choose the same dense parameters θ in RoSeViT and set the output projection of the structured route to zero, so $S_\psi(h) = 0$ for all h. The RoSeViT block then computes exactly D_θ, and repeating this argument for every block gives

$$\mathcal{F}_{\text{dense}} \subseteq \mathcal{F}_{\text{rose}}.$$

Second, we compute the entropy of the two rule routes. Let $\mathcal{Z} = [0,1]^d$. For a fixed task input, define

$$\Phi_i(z) = \phi(x_{\mathcal{N}(i)}, p_i, z), \qquad i = 1, \dots, L.$$

Assume each Φ_i is bi-Lipschitz in the instruction variable with constants $0 < a \le b$:

$$a\|z - z'\|_2 \le \|\Phi_i(z) - \Phi_i(z')\|_2 \le b\|z - z'\|_2.$$

Consider two rule routes. A site-wise patch-token route may assign an independent rule state to each site,

$$\mathcal{F}_{\text{site}} = \left\{ F_{z_1,\dots,z_L} : F_i = \Phi_i(z_i),\ z_i \in \mathcal{Z} \right\},$$

whereas a controller route uses one shared rule variable available to all sites,

$$\mathcal{F}_{\text{ctrl}} = \{F_z : F_i = \Phi_i(z),\ z \in \mathcal{Z}\}.$$

Under the site-wise metric

$$d_\infty(F, G) = \max_i \|F_i - G_i\|_2,$$

we prove

$$\log N(\epsilon, \mathcal{F}_{\text{ctrl}}, d_\infty) = \Theta\left(d \log \frac{1}{\epsilon}\right), \qquad \log N(\epsilon, \mathcal{F}_{\text{site}}, d_\infty) = \Theta\left(Ld \log \frac{1}{\epsilon}\right).$$

The shared-rule family is parameterized by a single instruction variable $z \in [0,1]^d$. Since every Φ_i is b-Lipschitz, an ϵ/b-cover of $[0,1]^d$ induces an ϵ-cover of $\mathcal{F}_{\text{ctrl}}$. Conversely, since each Φ_i is a-Lipschitz from below, an ϵ/a-packing of the unit cube induces an ϵ-packing of $\mathcal{F}_{\text{ctrl}}$. Standard covering and packing bounds for the d-dimensional cube therefore give

$$\log N(\epsilon, \mathcal{F}_{\text{ctrl}}, d_\infty) = \Theta\left(d \log \frac{1}{\epsilon}\right).$$

For the site-wise family, the parameters are $(z_1, \ldots, z_L) \in [0,1]^{Ld}$. Under d_∞, the induced distance is equivalent up to constants to $\max_i \|z_i - z_i'\|_2$. Thus a cover is obtained by independently covering each of the L copies of $[0,1]^d$, and a packing is obtained by independently packing each copy. This gives

$$\log N(\epsilon, \mathcal{F}_{\text{site}}, d_\infty) = \Theta\left(Ld \log \frac{1}{\epsilon}\right).$$

Third, we compute available solution-route entropy. The dense single-state ViT has the route set

$$\mathfrak{R}_{\text{dense}} = \{\mathcal{F}_{\text{site}}\},$$

so

$$\mathcal{H}_\epsilon(\mathfrak{R}_{\text{dense}}) = \log N(\epsilon, \mathcal{F}_{\text{site}}, d_\infty) = \Theta\left(Ld \log \frac{1}{\epsilon}\right).$$

RoSeViT keeps this dense route and adds the controller route:

$$\mathfrak{R}_{\text{rose}} = \{\mathcal{F}_{\text{site}}, \mathcal{F}_{\text{ctrl}}\}.$$

Therefore

$$\mathcal{H}_\epsilon(\mathfrak{R}_{\text{rose}}) = \min\{\log N(\epsilon, \mathcal{F}_{\text{site}}, d_\infty), \log N(\epsilon, \mathcal{F}_{\text{ctrl}}, d_\infty)\} = \Theta\left(d \log \frac{1}{\epsilon}\right).$$

For $L > 1$, adding the controller route decreases the available solution-route entropy by the workspace-size factor L. Together with the dense-route inclusion above, this proves Theorem 1.

Collapse condition. The entropy decrease above comes from adding $\mathcal{F}_{\text{ctrl}}$, whose free rule variable is shared across all workspace sites. A two-channel model whose workspace can freely store $z_1, \ldots, z_L$ leaves the site-wise route available as the effective rule route. The workspace stream can carry the rule independently at each site, the high-level channel can be bypassed, or both channels can redundantly represent the same rule information. In all such cases, the model has no guaranteed solution-route entropy decrease beyond $\Theta(Ld \log(1/\epsilon))$. An enforced bottleneck is therefore required for a two-channel design to realize the controller-route entropy reduction.

B Controller-Mediated Attention

Proposition 1 (Controller-mediated attention is globally low rank)**.** *Consider the structured RoSeViT pass with P controller tokens and L workspace tokens. Ignoring nonlinearities and attention-head constants, write the controller and workspace updates as*

$$C' = A_{CC}C + A_{CW}W, \qquad W' = A_{WC}C' + A_{WW}^{\text{loc}}W.$$

Then the nonlocal workspace-to-workspace influence factors as

$$A_{WC}A_{CW},$$

and therefore has rank at most P. A dense attention layer can realize a generic workspace-to-workspace influence matrix of rank up to L.

If each workspace token attends to k local neighbors in addition to the controller and itself, the structured pass uses

$$E_{\text{role}} = P(P+L) + L(P+k+1)$$

attention edges, while dense attention over the same tokens uses

$$E_{\text{dense}} = (P+L)^2.$$

Proof. Substituting the controller update into the workspace update gives

$$W' = A_{WC}A_{CC}C + A_{WC}A_{CW}W + A^{\text{loc}}_{WW}W.$$

The term $A^{\text{loc}}_{WW}W$ contains only local workspace interactions. The nonlocal workspace-mediated term is $A_{WC}A_{CW}W$. Since $A_{WC} \in \mathbb{R}^{L\times P}$ and $A_{CW} \in \mathbb{R}^{P\times L}$, the rank of their product is at most P. Dense attention imposes no such factorization and can represent a generic $L \times L$ interaction matrix.

The edge count follows from the attention mask. Controller tokens attend to all $P + L$ tokens, giving $P(P+L)$ edges. Each workspace token attends to the P controller tokens, itself, and k local neighbors, giving $L(P+k+1)$ edges. Dense attention allows all $P+L$ tokens to attend to all $P+L$ tokens, giving $(P+L)^2$ edges. □

C Experimental Details

C.1 ARC protocol and data

We follow the VARC two-stage protocol. Offline training learns one shared model across ARC-1 training tasks using the task demonstration sets. Following VARC, each ARC-1 training task is expanded with 1000 RE-ARC demonstrations [Hodel, 2024]. The inference examples of ARC-1 training tasks are withheld from offline training and used only for validation. ARC-2 tasks are not used in offline pretraining.

At test time, each unseen task is adapted from its own demonstration pairs only. The adapted model is then evaluated on the query inputs. We report pass@2 accuracy: a task is solved when either of the two submitted grids exactly matches the ground-truth output. Reported RoSeViT accuracies in the main tables are means over 4 seeds.

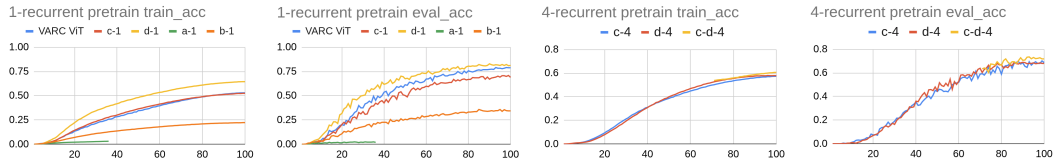
C.2 Shared visual configuration

All RoSeViT variants use the VARC visual frontend and output projection. Each grid is embedded into a 64×64 augmented canvas, tokenized with non-overlapping 2×2 patches, embedded into dimension 512, and processed by a depth-10 encoder. The controller contains one task token and one context token per demonstration pair. During offline training, each training task has a learned task token. During test-time adaptation, a fresh task token is initialized for the unseen task and optimized with the model.

C.3 Architecture variants

We denote each model as `arch-`r, where `arch`$\in \{a, b, c, d\}$ is the block design in Figure 4 and r is the number of recurrent refinement steps. The four block designs are:

- `a`: no dense visual pass; controller tokens attend globally; workspace tokens attend only to controller tokens.
- `b`: no dense visual pass; controller tokens attend globally; workspace tokens attend to controller tokens and a local 3×3 workspace neighborhood.



a) ablation study training accuracy b) ablation study evaluation accuracy c) best model training accuracy d) best model evaluation accuracy

Figure 7: Offline training and validation curves for representative RoSeViT block variants. Variants a/b enforce a strict controller bottleneck from the start and are too restrictive for ARC visual modeling. Variants c/d keep dense visual feature formation and then impose the controller/workspace route; d gives the strongest one-step backbone. For four recurrent steps, staged c-d training is most stable.

- `c`: dense visual self-attention followed by the structured pass; in the structured pass, workspace tokens attend only to controller tokens.
- `d`: dense visual self-attention followed by the structured pass; in the structured pass, workspace tokens attend to controller tokens and a local 3×3 workspace neighborhood.

Figure 7 records the architecture search evidence behind the main-text ablation. Variants a/b show that a bottleneck alone is too restrictive for learning ARC visual features. Variant c restores dense feature formation before the structured route, and variant d adds local workspace repair capacity. For $r = 4$, training d end-to-end was less stable in our runs; the best variant, `c-d-4`, first trains architecture c for 100 epochs and then enables the local workspace neighborhood of architecture d for 30 additional epochs.

C.4 Offline objectives

The base offline objective is output-grid cross-entropy after inverse augmentation and crop to the original grid. Recurrent models may also use a mid-layer cross-entropy loss for deep supervision [Lee et al., 2015]. In our implementation, $r \in \{1, 2\}$ models are trained without mid-loss, while the deeper recurrent setting uses mid-loss during offline training.

C.5 Test-time adaptation

For each unseen task, test-time training adapts the pretrained model using only that task's demonstrations, following the TTT paradigm [Akyürek et al., 2025, Sun et al., 2020]. Each adaptation epoch samples stochastic task-preserving augmentations of the demonstrations, including the rotation, translation, scale, and resolution augmentations used by VARC. Unless otherwise stated, adaptation updates the model end-to-end, including the fresh task token, and the optimizer state is reset at the start of each task.

We use three adaptation settings:

- `TTT1`: the VARC default adaptation setting, with 100 epochs and standard output-grid cross-entropy.
- `TTT2`: a longer, smaller-step adaptation setting with a learning rate reduced by $3\times$, 300 epochs, and early termination after a 3-epoch streak of 100% demonstration accuracy.
- `TTT3`: `TTT2` plus adaptation-only auxiliary objectives for exact-grid correction: a grid-level correctness head, pixel-level error emphasis for sparse residual mistakes in the spirit of focal loss [Lin et al., 2017], and reward shaping that down-weights output cross-entropy and disables mid-loss after an exactly correct demonstration prediction is reached.

C.6 Inference and attention visualization

Inference follows VARC multi-view voting. For each adapted task and query input, we evaluate 510 augmented views, invert each prediction back to the original grid coordinates, consolidate predictions by majority vote, and submit the two most frequent grids as the pass@2 outputs.

Figure 5 compares RoSeViT d-1-TTT1 and VARC-ViT under the same training and testing setting. For visualization, the selected ARC example uses ratio-2 upscaling so each original grid cell occupies exactly one patch; this makes the patch-level attention map align with the pixel grid.

D Parameter-Matched Control, Controller-Route Experiments, and Cost Measurement

The experiments in this appendix were run with the official VARC codebase and protocol on an 8×H100 node; adaptation and scoring conventions are those of Section 5.

D.1 Parameter-matched dense control

The width-matched VARC-ViT keeps the default depth and head count and widens the embedding from 512 to 640 (widths are constrained to multiples of 64 by the rotary-embedding and kernel divisibility of the codebase); the residual capacity is placed in the MLP width (`--mlp-dim` 720). The VARC parameter counter reports 28.00M, against 28.43M for RoSeViT d-1. Everything else follows the official scripts: 100 offline epochs, VARC default adaptation (TTT1), 510-view voting, pass@2. The reported 53.6 ± 1.0 is the mean and standard deviation over 8 adaptation seeds (range 52.1–54.9); offline training curves are smooth and reach 77.4% offline exact match at the final epoch, so the control is not undertrained.

D.2 Controller ablation

Controllers of RoSeViT d-1 are one task token and four context tokens. An ablated inference zeroes the selected controller embeddings at their source and leaves the weights, attention masks, dense pass, and local workspace neighborhoods untouched. The offline protocol scores exact match on the held-out test pairs of the ARC-1 training tasks, the same pairs used for offline validation, under the training-time evaluation protocol. Conditions: *all conditioning zeroed* removes the task and context tokens (for the dense control, which has a task token but no context tokens, it removes the task token); *batch-mean controllers* replaces each controller with its mean over the evaluation batch; *wrong-task controllers* swaps in the controllers of a different training task; *task token only* and *context tokens only* zero one carrier at a time.

The post-TTT protocol uses the 400 unseen ARC-1 evaluation tasks under the full adaptation and voting pipeline. For each of two adaptation seeds, the adapted weights of each task are used twice, once with intact controllers and once with the controllers (RoSeViT) or the task token (dense control) zeroed, so the intact and ablated pass@2 values of a seed come from the same weights. The intact values (56.25/56.25 for RoSeViT d-1, 52.9/54.9 for the dense control) are consistent with Table 1.

D.3 Rule probe

The probe decodes the demonstration-defined transformation rule (one of the ARC-1 training rules; chance 0.25%) from frozen token states of RoSeViT d-1 with a linear classifier, separately from the context tokens and from the workspace tokens at layers 1, 3, 6, and 10. The task token is excluded from the probe features to avoid a circular readout of task identity. Because naive decodability saturates on appearance confounds (the input grid of a training task identifies the task), the probe uses a dissociation design in which the rule label must be recovered from states that are not trivially aligned with appearance; the reported accuracies are on held-out pairs. Rule information is about twice as decodable from the context tokens as from the workspace tokens at every depth (64.1/53.8/44.0/30.1 versus 33.3/27.7/24.2/16.8).

D.4 Compute and latency measurement

Forward cost was measured on the as-published implementations with a batch of 8 canvases on one H100 GPU: dense VARC-ViT 18M, 53.8 GFLOPs and 20.7 ms; the width-matched dense 28M control, 79.4 GFLOPs and 22.2 ms; RoSeViT d-1, 97.3 GFLOPs and 46.1 ms. The theoretical figure of 75.9 GFLOPs for RoSeViT d-1 counts the structured pass at the sparse edge set of Section 4 (each workspace token attends to at most 14 tokens). Our implementation does not realize this sparsity:

no kernel for this access pattern was developed in this work, and the structured pass is executed as dense attention under a boolean mask, which is why actual FLOPs and latency exceed the matched dense control by about 23% and $2.1\times$. The difference between the theoretical and actual cost is an engineering gap, not a property of the architecture, but the paper makes no efficiency claim.

E Preliminary Transfer Evidence: Instruction-Conditioned Maze Suite

To test whether the architectural advantage appears on a second visual transformation task family, we built an instruction-conditioned maze suite and compared parameter-matched RoSeViT and dense ViT backbones under the paper's protocol without any of the amplifiers: feedforward (no recurrence), offline training only, no test-time training, no voting, no ensembling, two seeds per arm. Training data are generated procedurally on the fly, so no result can be explained by memorization. We report this study as preliminary: the suite is new, two seeds were run, and four of the eight rules are unsolved by both models within the training budget.

Data. Each example is a 45×45 corridor maze on a 48×48 canvas (mirroring the 30-in-32 geometry of HRM Maze-Hard), generated by recursive backtracking with about 4% of interior walls knocked out to create loops. Start and goal are sampled so that the BFS shortest-path length is at least 45 (median about 83). The eight rules are all derived from the BFS distance maps d_s (from the start) and d_g (from the goal) on the same input (Figure 8): T0 shortest path (also scored by a strict validity checker: the predicted cells must form a connected path of optimal length); T1 wavefront from the start ($d_s \leq L/2$); T2 dead-ends (free cells with exactly one open neighbor, a purely local rule); T3 even-distance parity; T4 wavefront from the goal ($d_g \leq L/2$; the same geometry as T1 with the opposite anchor, a pure instruction discrimination); T5 far field (top-decile d_s); T6 distance mod 3; T7 geodesic set (all cells on any shortest path). Scoring is exact match of the full output grid on a fixed held-out set of 128 mazes $\times$ 8 rules.

Models and training. Both models are ViTs of width 512 with 8 heads and patch size 2 (a 24×24 grid of 576 workspace tokens). RoSeViT uses depth 8 (33.99M parameters); each block is a dense pass, a structured pass in which 5 controller tokens attend globally and grid tokens see the controllers and their 3×3 neighborhood, and an MLP, with zero-initialized gates. The dense ViT uses standard pre-LN blocks at depth 11 (35.02M parameters), the depth whose parameter count is nearest to RoSeViT's. The task identity is one learned embedding given to both models: RoSeViT receives it through its task token, the dense baseline as an extra token attended globally by all tokens. Both arms use AdamW with learning rate 3×10^{-4} and cosine decay, bf16, a global batch of 256, and 40k steps, trained on two H200 GPUs per arm.

Results. Table 4 reports the full emergence grid, exact match per rule every 5k steps for every arm and seed. Both models solve the sanity rule T3 immediately. On the local rule T2, RoSeViT is near 0.98 by 10k steps while the dense baseline stays at 0.00 until about 15k and needs 25–30k steps to master the same rule. On the global rules T0 and T7, RoSeViT's accuracy emerges at about 15k steps and climbs to 0.51/0.34 (T0, strict solve) and 0.53/0.38 (T7) at 40k, whereas the dense baseline reaches at most 0.02 within the same budget on either seed. T1, T4, T5, and T6 are beyond both models at this budget. The two architectures therefore climb the same difficulty ladder, with RoSeViT reaching each rung earlier; at a fixed budget this timing difference appears as large, sometimes total, performance gaps. For completeness we also evaluated the public HRM Maze-Hard-1k benchmark: feedforward models of both architectures score 0%, matching the HRM paper's own report that non-recurrent Transformers score 0% there (shortest paths of length 110 or more exceed feedforward computational depth), so that benchmark cannot discriminate between feedforward architectures by design; its 1k-sample training regime also rewards memorization under recurrent training. We therefore report the instruction-conditioned suite above as the primary maze evidence.

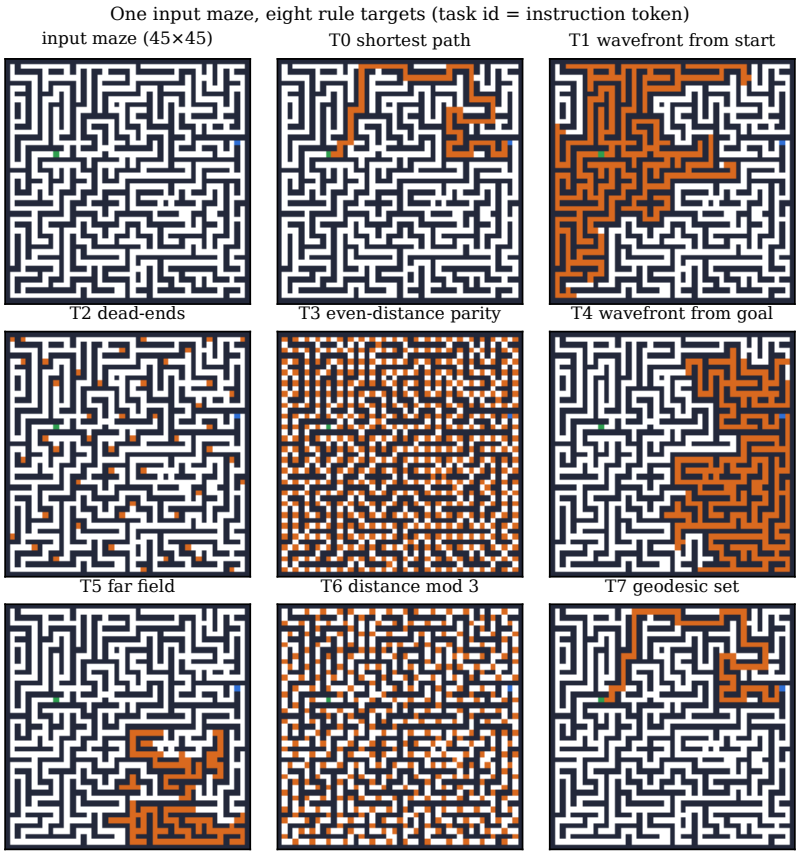


Figure 8: The maze suite: one input maze and its eight rule targets. The rule is given as a task id (an instruction token for RoSeViT; the same embedding as an extra globally attended token for the dense baseline), so both models must apply the selected rule to the same input.

Table 4: Maze suite emergence grid: exact-match accuracy on the held-out set per rule, arm, and seed, every 5k training steps (feedforward, offline, no test-time training, no voting). T0 is scored by the strict validity checker.

Rule	Arm (seed)	5k	10k	15k	20k	25k	30k	35k	40k
T0 shortest path	RoSeViT (42)	.00	.00	.02	.15	.29	.40	.47	.51
	RoSeViT (43)	.00	.00	.00	.06	.24	.31	.35	.34
	Dense (42)	.00	.00	.00	.00	.00	.00	.02	.02
	Dense (43)	.00	.00	.00	.00	.00	.00	.00	.00
T1 wavefront-S	RoSeViT (42)	.00	.00	.01	.01	.01	.02	.02	.05
	RoSeViT (43)	.00	.00	.00	.00	.01	.00	.02	.02
	Dense (42)	.00	.00	.00	.00	.00	.01	.01	.01
	Dense (43)	.00	.00	.00	.00	.00	.00	.00	.00
T2 dead-ends	RoSeViT (42)	.76	.98	.99	1.00	1.00	1.00	1.00	1.00
	RoSeViT (43)	.82	.97	1.00	1.00	1.00	1.00	1.00	1.00
	Dense (42)	.00	.00	.35	.88	1.00	.99	1.00	1.00
	Dense (43)	.00	.00	.00	.35	.91	.99	1.00	.99
T3 parity	RoSeViT (42)	1.00	1.00	.99	1.00	1.00	1.00	1.00	1.00
	RoSeViT (43)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Dense (42)	1.00	.99	1.00	1.00	1.00	1.00	1.00	1.00
	Dense (43)	.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
T4 wavefront-G	RoSeViT (42)	.00	.00	.00	.01	.00	.02	.02	.02
	RoSeViT (43)	.00	.00	.00	.00	.01	.01	.01	.02
	Dense (42)	.00	.00	.00	.00	.00	.00	.01	.02
	Dense (43)	.00	.00	.00	.00	.00	.00	.00	.00
T5 far field	RoSeViT (42)	.00	.00	.00	.01	.03	.05	.02	.02
	RoSeViT (43)	.00	.00	.00	.00	.00	.03	.02	.03
	Dense (42)	.00	.00	.01	.00	.00	.01	.00	.01
	Dense (43)	.00	.00	.00	.00	.00	.00	.00	.00
T6 mod 3	RoSeViT (42)	.00	.00	.00	.00	.00	.00	.00	.00
	RoSeViT (43)	.00	.00	.00	.00	.00	.00	.00	.00
	Dense (42)	.00	.00	.00	.00	.00	.00	.00	.00
	Dense (43)	.00	.00	.00	.00	.00	.00	.00	.00
T7 geodesic set	RoSeViT (42)	.00	.00	.03	.16	.30	.37	.49	.53
	RoSeViT (43)	.00	.00	.00	.08	.23	.29	.37	.38
	Dense (42)	.00	.00	.00	.00	.00	.00	.01	.02
	Dense (43)	.00	.00	.00	.00	.00	.00	.00	.00

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state the motivating theoretical result, the RoSeViT architecture, the controlled backbone comparison (with a parameter-matched control), and the full-system ARC results; Section 6 states the scope and limitations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Section 6 confines the claims to ARC-type shared-rule visual transformation reasoning, states the single-shared-rule assumption of the theorem, the preliminary status of the transfer evidence (Appendix E), the added compute and $2.1\times$ latency of our implementation, the dependence on the VARC adaptation/voting stack, and that rule information remains decodable from workspace tokens.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes].

   Justification: Theorem 1 states its assumptions, Proposition 1 states the attention factorization setting, and Appendix A and Appendix B provide the proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: Section 5 gives the main protocol, Appendix C gives training, adaptation, inference, and variant details, and the released repository (`https://github.com/lz7fd/RoSeViT`) gives reproduction commands.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: Code, requirements, training scripts, TTT scripts, analysis scripts, trained checkpoints, raw prediction archives, and a README with the reproduction workflow are released at `https://github.com/lz7fd/RoSeViT`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Section 5 specifies the protocol needed to interpret the results, and Appendix C gives data splits, model variants, offline objectives, TTT settings, and inference/voting details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

Justification: RoSeViT results are reported as means over 4 seeds, the parameter-matched dense control as mean and standard deviation over 8 adaptation seeds with its range (Section 5.1, Appendix D.1), the post-TTT ablation per seed (Table 2), the maze suite per seed (Table 4), key standard deviations in the main text for the strongest single model, and the released prediction archives allow the reported results to be recomputed.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: The reported experiments were run with 8 NVIDIA B200 GPUs; the released repository README gives per-script hardware and wall-clock time, and the scripts encode 8-GPU training and task-parallel TTT execution. The parameter-matched control, the controller ablation and probe, and the cost measurements (Appendix D) were run on an 8×H100 node, and the maze suite (Appendix E) on an 8×H200 node with two GPUs per arm.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The work uses public ARC-style benchmark data and standard model training/evaluation; we are not aware of any deviation from the NeurIPS Code of Ethics.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A].

    Justification: The work is foundational research on ARC visual reasoning and does not introduce a deployed system or application-specific societal intervention.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The work does not release a high-risk pre-trained language model, image generator, scraped dataset, or deployment-facing system requiring special safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: We cite and credit ARC-AGI-1/2, RE-ARC, and VARC. ARC-AGI-1/2 are Apache-2.0, RE-ARC is MIT, and the VARC paper/protocol is CC BY 4.0.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The released repository includes code, scripts, checkpoints, raw prediction archives, result visualizations, and a README documenting the directory layout, hardware, data layout, and reproduction workflow.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: This work does not involve human subjects, so IRB or equivalent approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core method and experiments do not use LLMs as an important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.