OFF-DIAGONAL ATLAS TRANSPORT

Anonymous authors
Paper under double-blind review

ABSTRACT

Next-token prediction determines what a language model must predict, but does not uniquely determine how task-relevant information is organized across token positions. Temporal representation objectives can shape this organization. However, matching the outputs of a learned projection across positions admits an easy shortcut: a persistent output direction can satisfy the objective without tracking hidden-state changes. We introduce Off-Diagonal Atlas Transport (OAT), a training-time regularizer that maps each hidden state through multiple learned projections and rewards temporal compatibility only across different projection indices. Because off-diagonal matching can itself become trivial when projections of the same state become nearly identical, OAT also suppresses positive overlap among them. Across three instruction-tuned language-model families and three low-data fine-tuning tasks, OAT improves mean accuracy over next-token prediction in all nine backbone–task combinations, with its clearest gains on mathematical reasoning. Ablations show that allowing same-index temporal matches or removing cross-projection redundancy suppression makes the auxiliary objective easy to satisfy. Yet the downstream benefit largely disappears under these changes.

1 INTRODUCTION

Transformer language models solve tasks that require information to be formed, updated, and reused across token positions. Next-token training can produce structured internal states in the residual stream (Shai et al., 2024), yet models trained on the same task can learn different state-tracking mechanisms (Li et al., 2025). How information is organized as computation unfolds therefore remains a meaningful target for regularization. In particular, when useful information becomes available can matter to subsequent computation (Lepori et al., 2025).

Temporal representation objectives provide one way to shape this organization. Semantic Tube Prediction (STP), for example, encourages local linearity in hidden-state trajectories (Huang et al., 2026b). We seek a more flexible form of temporal regularization that encourages compatibility across positions without requiring the hidden-state trajectory itself to be locally straight. To this end, we consider matching the outputs of a learned readout across token positions.

Such matching, however, admits a simple shortcut: a persistent output direction can satisfy the temporal objective without tracking hidden-state changes. We introduce **Off-Diagonal Atlas Transport (OAT)**, which maps each hidden state through multiple learned projections and rewards temporal matches only between different projection indices. Because different projections can still make this matching trivial by producing nearly identical views, OAT also suppresses positive overlap among projections of the same state. The projections are trained jointly with the language model and discarded at inference, leaving the autoregressive inference graph unchanged.

Across three instruction-tuned model families and three low-data fine-tuning tasks, OAT improves mean accuracy over next-token prediction in all nine backbone–task combinations. Its clearest advantage over STP appears on mathematical reasoning, whereas STP remains stronger on the natural-language-to-regular-expression tasks. Structural ablations confirm the role of each component: allowing same-index matches or removing redundancy suppression makes the auxiliary objective easy to satisfy, but largely eliminates the downstream benefit. To better understand the scope and limits of this regularization, we examine its behavior across data fractions, model scales, and frame counts, and we evaluate whether OAT can also support training from random initialization.

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

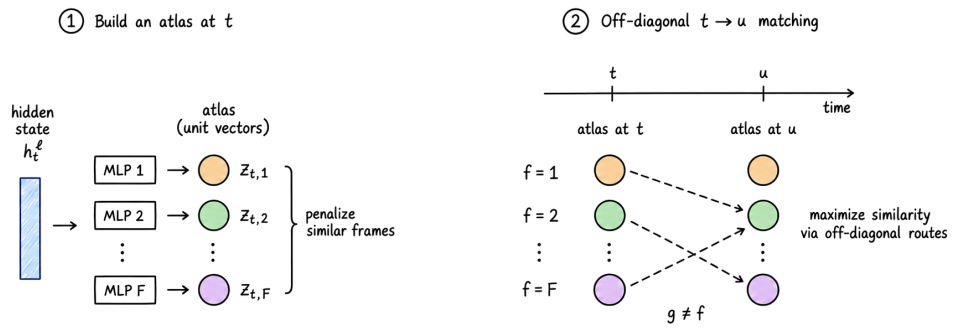

Figure 1: Overview of Off-Diagonal Atlas Transport. A hidden state at a sampled Transformer layer is mapped through multiple learned projections. OAT excludes same-index source–target pairs, matches each projected source representation to its most compatible different-index future representation, and suppresses positive redundancy among projections of the same source state. Future projected representations are stop-gradient targets.

2 Off-Diagonal Atlas Transport

OAT regularizes temporal relationships among multiple learned projections of Transformer hidden states, as illustrated in Figure 1. We use *transport* to refer to the resulting off-diagonal temporal coupling rather than an explicitly parameterized transition operator.

2.1 Multi-Frame Atlas

Consider a decoder-only Transformer with hidden dimension D, and let $h_t^\ell \in \mathbb{R}^D$ denote the residual-stream representation at token position t after block ℓ. OAT defines F learned projection frames $P_1, \ldots, P_F$, each mapping the same hidden state into an auxiliary space of dimension d_{proj}:

$$z_{t,f}^\ell = \text{normalize}\left(P_f\left(\text{LN}(h_t^\ell)\right)\right), \qquad f = 1, \ldots, F. \tag{1}$$

Each P_f is a lightweight two-layer MLP, and each output is normalized to unit length. We call P_f a *frame*, $z_{t,f}^\ell$ its corresponding *view*, and the collection of frames an *atlas*. The projection parameters are shared across token positions, examples, and the layers on which OAT is applied. The frames are learned readouts of a common hidden state rather than separate model branches, and no semantic role is assigned to a frame index.

2.2 Off-Diagonal Transport

OAT samples source and target positions from valid content tokens. For each example, source positions are sampled without replacement from positions that admit at least K eligible future targets. A future position u is eligible for source t when $u - t > \delta$, where δ is a minimum temporal gap; K eligible targets are then sampled uniformly without replacement.

For a sampled pair (t, u), define the cross-frame similarity $C_{fg}(t, u) = (z_{t,f}^\ell)^\top \text{sg}(z_{u,g}^\ell)$, where sg denotes stop-gradient. A naive temporal objective could allow $g = f$, in which case each frame can retain a fixed identity and its view can match itself across positions. OAT instead masks the diagonal and scores each source view by its strongest different-index target view. Averaging over frames and the K future targets gives

$$P(t) = \frac{1}{K}\sum_{i=1}^{K}\frac{1}{F}\sum_{f=1}^{F}\max_{g \neq f} C_{fg}(t, u_i). \tag{2}$$

Each source view selects its best different-index target independently, so multiple source views may select the same target frame.

2.3 Suppressing Trivial Solutions

Diagonal exclusion alone is insufficient: if different views of the same state become highly aligned, changing the matched frame index need not change the matched representation. OAT therefore measures positive directional redundancy among views of the same state:

$$N(t) = \frac{1}{F} \sum_{f=1}^{F} \max_{g \neq f} \left[\left(z_{t,f}^{\ell} \right)^{\top} z_{t,g}^{\ell} \right]_{+}, \tag{3}$$

where $[x]_+ = \max(x, 0)$. This term penalizes the strongest positive overlap seen by each frame but does not push already orthogonal or anti-correlated pairs farther apart.

The auxiliary loss compares off-diagonal temporal compatibility with within-state redundancy:

$$\mathcal{L}_{\mathrm{OAT}} = \mathbb{E}_t \left[\mathrm{softplus} \left(\frac{N(t) - P(t)}{\tau} \right) \right], \tag{4}$$

where $\tau > 0$ is a temperature. Minimizing this loss encourages each source's off-diagonal temporal compatibility to exceed its within-state redundancy. The objective thus couples temporal matching and redundancy suppression through their difference.

2.4 Training and Inference

The Transformer and projection frames are trained jointly with $\mathcal{L}_{\mathrm{main}} = \mathcal{L}_{\mathrm{NTP}} + \lambda \mathcal{L}_{\mathrm{OAT}}$, where $\mathcal{L}_{\mathrm{NTP}}$ is the ordinary autoregressive language-modeling loss. For each application of the OAT loss during training, one layer ℓ is sampled from a specified set of intermediate layers, with the same projection frames shared across that set. The OAT coefficient is ramped from zero early in training to its target value. At inference time, the projection frames and auxiliary objective are removed; generation uses the original language-model head and autoregressive architecture. In a wall-clock benchmark on Llama-3.2-1B with GSM8K, OAT increases steady-state training time by approximately 4% while adding no inference-time computation.

3 Experiments

We evaluate OAT across several training regimes. We begin with full-parameter fine-tuning of pre-trained instruction-tuned models in low-data settings, comparing against next-token prediction and STP across three model families. To examine where these gains persist, we vary the training-data fraction and evaluate model scale up to 3B parameters. We then ablate the two structural constraints and assess sensitivity to the number of projection frames. Finally, we test whether the same regularizing principle can support training from random initialization.

3.1 Fine-Tuning Setup

Models and tasks. Our fine-tuning evaluation follows the task and data-efficiency setting used by Semantic Tube Prediction (STP) (Huang et al., 2026b) where applicable, while re-running all methods under a common generation-supervision protocol. The main comparison uses three decoder-only instruction-tuned backbones: Llama-3.2-1B-Instruct, OLMo-2-0425-1B-Instruct, and Qwen2.5-0.5B-Instruct (Meta, 2024; Allen Institute for AI, 2025; Yang et al., 2024). We evaluate natural-language-to-regular-expression generation on NL-RX-SYNTH and NL-RX-TURK (Locascio et al., 2016), using the datasets released with STP, and mathematical reasoning on GSM8K (Cobbe et al., 2021). Models fine-tuned on GSM8K are additionally evaluated on GSM1K (Zhang et al., 2024) without further training or hyperparameter adjustment. We also run a full-data GSM8K scale check with Llama-3.2-3B-Instruct. For SYNTH and TURK, evaluation requires an exact match to the target string. For GSM8K and GSM1K, we extract the final answer following the standard #### delimiter and compare the extracted answer exactly.

Training protocol. All fine-tuning results are means and sample standard deviations over five matched seeds, 42–46. Unless stated otherwise, the main low-data setting uses $1/16$ of the available training set. To separate unique-data diversity from the number of training presentations, we keep

Table 1: Accuracy (%) at $1/16$ training data. Values are mean $\pm$ sample standard deviation over five matched seeds. GSM1K is evaluated without retuning using the corresponding GSM8K checkpoint. Bold indicates the highest mean within each backbone and evaluation column.

Backbone	Method	SYNTH	TURK	GSM8K	GSM1K
Llama-3.2-1B	NTP	66.82 ± 0.44	24.17 ± 0.65	30.67 ± 1.21	27.29 ± 0.49
	STP	$\mathbf{68.55 \pm 0.49}$	$\mathbf{26.53 \pm 0.58}$	31.46 ± 0.71	28.22 ± 1.07
	OAT	67.76 ± 0.66	25.24 ± 0.27	$\mathbf{32.43 \pm 1.12}$	$\mathbf{28.78 \pm 0.83}$
OLMo-2-1B	NTP	62.89 ± 0.33	18.92 ± 0.48	37.57 ± 0.57	45.38 ± 0.72
	STP	$\mathbf{63.62 \pm 0.32}$	$\mathbf{20.18 \pm 0.51}$	37.83 ± 0.52	45.66 ± 0.47
	OAT	63.17 ± 0.23	19.35 ± 0.37	$\mathbf{39.21 \pm 0.45}$	$\mathbf{45.99 \pm 1.15}$
Qwen2.5-0.5B	NTP	63.58 ± 0.41	20.27 ± 0.38	22.55 ± 1.12	21.29 ± 0.74
	STP	$\mathbf{65.47 \pm 0.33}$	$\mathbf{21.94 \pm 0.26}$	22.09 ± 0.87	21.88 ± 0.78
	OAT	64.18 ± 0.18	20.74 ± 0.26	$\mathbf{25.41 \pm 1.04}$	$\mathbf{23.52 \pm 1.39}$

the presentation budget approximately fixed across data fractions: full-data runs use four epochs, $1/4$-data runs use 16 epochs, and $1/16$-data runs use 64 epochs. We use AdamW (Loshchilov & Hutter, 2019) with learning rate 2×10^{-5}, 5% linear warmup followed by cosine decay to 0.1 of the peak learning rate, weight decay 0.01, $\beta = (0.9, 0.98)$, gradient clipping at norm 1, and no label smoothing.

Baselines and hyperparameter selection. We compare ordinary next-token prediction (NTP), STP (Huang et al., 2026b), and OAT. STP is implemented as the public random-span semantic-trajectory objective on the final Transformer block, with one sampled span, no length adjustment, and no linear predictor. For both auxiliary objectives, coefficients are selected by mean test accuracy over five seeds on Llama at $1/16$ data, using SYNTH for RX and GSM8K for math under a common factor-of-two sweep protocol. TURK and GSM1K inherit the RX and math coefficients, respectively. This yields $\lambda_{\mathrm{OAT}} = 0.02$ for RX and 0.04 for math, and $\lambda_{\mathrm{STP}} = 0.32$ for RX and 0.02 for math. The selected coefficients are transferred unchanged to OLMo, Qwen, and the Llama-3.2-3B scale check, with no backbone-specific retuning. Selection details and the OAT sweep are reported in Appendix A.

OAT is ramped from zero to its selected coefficient between 5% and 25% of training, whereas STP uses its selected coefficient from the start, matching its standard setting (Huang et al., 2026b). All main comparisons supervise the assistant end-of-turn (EOT) token because otherwise exact-match evaluation is strongly confounded by termination behavior.[1]

OAT configuration. Unless otherwise stated, OAT fine-tuning runs use $F = 16$ frames with $d_{\mathrm{proj}} = 128$, up to 24 sampled source positions per example, $K = 2$ future targets per source, and temperature $\tau = 0.1$. The choice $F = 16$ was fixed before the frame-count sensitivity experiment in Section 3.4. For each application of OAT, we sample one intermediate layer from the 0-indexed block sets $\{6, 7, 8, 9\}$ for Llama-3.2-1B and OLMo, $\{10, 11, 12, 13\}$ for Qwen, and $\{12, 13, 14, 15\}$ for Llama-3.2-3B. The minimum temporal gap is $\delta = 5$ on SYNTH and TURK and $\delta = 10$ on GSM8K. OAT samples from user and assistant content tokens while excluding structural chat tokens.

3.2 Main Fine-Tuning Results

Table 1 reports the main $1/16$-data comparison. OAT improves mean accuracy over NTP in all nine backbone–training-task combinations: SYNTH, TURK, and GSM8K for each of the three models.

On GSM8K, OAT improves over NTP by 1.76, 1.64, and 2.87 percentage points for Llama, OLMo, and Qwen, respectively, and exceeds STP by 0.97, 1.38, and 3.32 points. The corresponding GSM1K

[1]The public STP preprocessing code includes an assistant-content-only label-construction path that excludes structural tokens. In our corresponding Llama/SYNTH $1/16$-data control experiment, removing EOT supervision lowers NTP from 66.82 ± 0.44 to 29.51 ± 1.82; many failures generate the correct target as a prefix and then continue. Our EOT-supervised Llama/SYNTH NTP baseline is also higher than the corresponding Llama/SYNTH $1/16$-data STP result reported by Huang et al. (2026b) under their original supervision protocol, illustrating that absolute scores across these protocols are not directly comparable. We therefore re-run NTP, STP, and OAT with identical EOT supervision.

Table 2: Llama accuracy (%) as a function of training-data fraction under an approximately fixed presentation budget. Values are mean ± sample standard deviation over five seeds. GSM1K uses checkpoints trained on the corresponding GSM8K fraction. Bold indicates the highest mean within each row.

Evaluation	Fraction	NTP	STP	OAT
SYNTH	1/16	66.82 ± 0.44	$\mathbf{68.55 \pm 0.49}$	67.76 ± 0.66
	1/4	82.07 ± 0.30	$\mathbf{83.59 \pm 0.25}$	82.33 ± 0.27
	Full	$\mathbf{86.77 \pm 0.29}$	85.48 ± 0.57	86.62 ± 0.38
TURK	1/16	24.17 ± 0.65	$\mathbf{26.53 \pm 0.58}$	25.24 ± 0.27
	1/4	33.16 ± 0.81	$\mathbf{34.46 \pm 0.38}$	33.60 ± 0.29
	Full	43.17 ± 0.48	43.19 ± 0.28	$\mathbf{43.22 \pm 0.26}$
GSM8K	1/16	30.67 ± 1.21	31.46 ± 0.71	$\mathbf{32.43 \pm 1.12}$
	1/4	27.98 ± 0.54	28.04 ± 0.44	$\mathbf{28.10 \pm 0.98}$
	Full	41.47 ± 1.10	42.02 ± 0.66	$\mathbf{42.23 \pm 0.71}$
GSM1K	1/16	27.29 ± 0.49	28.22 ± 1.07	$\mathbf{28.78 \pm 0.83}$
	1/4	25.61 ± 0.46	25.08 ± 0.79	$\mathbf{26.27 \pm 0.89}$
	Full	$\mathbf{37.69 \pm 0.48}$	37.43 ± 0.21	37.36 ± 0.40

Table 3: Full-data Llama-3.2-3B scale check. Values are accuracy (%), mean ± sample standard deviation over five matched seeds. The OAT and STP math coefficients are transferred without 3B-specific retuning. GSM1K is evaluated from the corresponding GSM8K checkpoint.

Method	GSM8K	GSM1K
NTP	64.25 ± 1.12	61.53 ± 0.46
STP	64.26 ± 2.11	61.06 ± 1.06
OAT	$\mathbf{64.50 \pm 0.67}$	$\mathbf{61.56 \pm 1.02}$

gains over NTP are 1.49, 0.61, and 2.22 points, indicating that part of the improvement transfers to a separately authored GSM-style benchmark.

On SYNTH and TURK, OAT improves over NTP for every backbone, but STP consistently yields larger gains. These gains over NTP are obtained with the Llama-selected OAT coefficients transferred to OLMo and Qwen without retuning.

3.3 Dependence on Training-Data Fraction

We next vary the amount of unique training data on Llama while holding the number of example presentations approximately fixed. This changes the balance between diversity and repeated exposure rather than simply increasing optimization compute. Table 2 reports NTP, STP, and OAT at $1/16$, $1/4$, and full data.

On the RX tasks, OAT's gains over NTP are largest at $1/16$ data and diminish as the data fraction increases. At full data, OAT slightly trails NTP on SYNTH and is nearly tied on TURK. STP provides larger low-data gains but falls below NTP on full-data SYNTH. The math results are nonmonotonic: OAT improves GSM8K at all three fractions, with the smallest gain at $1/4$ data, whereas its GSM1K gain disappears at full data. Thus, the dependence on data fraction differs across task families.

Full-data scale check. We repeat the full-data GSM8K comparison with Llama-3.2-3B-Instruct, transferring the math coefficients without retuning. In Table 3, OAT achieves the highest mean GSM8K accuracy (64.50 ± 0.67), but its margin over NTP and STP is small and the advantage is not consistent across seeds. On GSM1K, OAT is essentially tied with NTP (61.56 versus 61.53). These results show only a modest in-domain mean improvement in this larger-model, full-data setting.

Table 4: Structural OAT ablations on Llama at $1/16$ data. Values are accuracy (%), mean $\pm$ sample standard deviation over five matched seeds. GSM1K is evaluated from the GSM8K checkpoint.

Configuration	SYNTH	GSM8K	GSM1K
NTP	66.82 ± 0.44	30.67 ± 1.21	27.29 ± 0.49
OAT	$\mathbf{67.76 \pm 0.66}$	$\mathbf{32.43 \pm 1.12}$	$\mathbf{28.78 \pm 0.83}$
Diagonal allowed	66.76 ± 0.55	30.16 ± 1.00	27.63 ± 0.66
No redundancy	66.86 ± 0.82	30.52 ± 1.11	27.35 ± 0.29

Table 5: Late-training GSM8K diagnostics for the structural ablations. $\mathcal{L}_{\text{aux}}$ and P_{used} are the auxiliary loss and temporal score used by each configuration; P_{used} includes same-index matches only for Diagonal allowed. P_{diag} is same-index similarity, N is within-state redundancy, and occupancy is the maximum fraction of source views selecting one target-frame index.

Configuration	$\mathcal{L}_{\text{aux}}$	P_{used}	P_{diag}	N	Occupancy
OAT	0.2407	0.184	0.591	0.030	0.156
Diagonal allowed	0.0005	0.796	0.796	0.033	0.0625
No redundancy	0.0008	0.716	0.739	0.715	0.187

3.4 Why Both Structural Constraints Matter

OAT uses diagonal exclusion and within-state redundancy suppression to address persistent frame identity and high alignment among views of the same state. We ablate these constraints independently on Llama at $1/16$ data while keeping the selected coefficient and all other training settings fixed. In the *diagonal-allowed* ablation, Equation 2 uses $\max_g C_{fg}$ rather than $\max_{g \neq f} C_{fg}$. In the *no-redundancy* ablation, $N(t)$ is removed from Equation 4 but remains measured as a diagnostic. Table 4 reports downstream accuracy for the two ablations.

Both ablations largely eliminate OAT's downstream benefit. OAT exceeds each ablation for all five matched seeds on SYNTH, GSM8K, and GSM1K. The auxiliary diagnostics in Table 5 distinguish the two failure patterns.

With diagonal matches allowed, the temporal score is nearly identical to same-index similarity, while redundancy remains low and occupancy reaches $1/F = 0.0625$. This pattern is consistent with distinct views matching their own frame identities across positions. Removing redundancy suppression produces a different pattern: within-state redundancy rises from 0.030 to 0.715, indicating strong alignment among views even though temporal matches remain off-diagonal.

In both cases, the modified auxiliary loss approaches zero, yet downstream accuracy returns to the NTP baseline range. Together, these results support the joint use of diagonal exclusion and redundancy suppression in the evaluated setting. Appendix B further examines whether OAT's matching behavior can be explained by a fixed position-conditioned routing rule.

Number of frames. We test frame-count sensitivity on Llama/GSM8K at $1/16$ data by reducing F from the previously fixed value of 16 to 2, 4, and 8, with all other OAT settings unchanged. Mean accuracy increases monotonically with F on both GSM8K and GSM1K (Table 6). At $F = 2$, each source frame has only one admissible target frame, so matching reduces to a fixed cross-frame pairing. This configuration performs essentially at the NTP baseline on GSM8K.

The late-training OAT loss increases from 0.111 at $F = 2$ to 0.150, 0.181, and 0.241 at $F = 4$, 8, and 16, respectively, alongside the increase in downstream accuracy. Varying F changes both the number of admissible target-frame choices and the number of projector parameters, so we do not attribute the accuracy trend to a single mechanism.

3.5 Beyond Pre-Trained Fine-Tuning

We next examine training from random initialization. In preliminary runs, vanilla OAT rapidly finds low-loss solutions that can switch target-frame identity at adjacent target positions without improving language modeling. We therefore use a *stabilized OAT* variant with a smaller projector

Table 6: Sensitivity to the number of OAT frames on Llama-3.2-1B at $1/16$ GSM8K training data. The $F = 16$ setting was fixed before this sensitivity experiment; all other OAT settings are unchanged. Values are accuracy (%), mean $\pm$ sample standard deviation over five matched seeds.

Configuration	GSM8K	GSM1K
NTP	30.67 ± 1.21	27.29 ± 0.49
$F = 2$	30.72 ± 1.05	27.47 ± 0.49
$F = 4$	31.39 ± 0.14	27.52 ± 1.03
$F = 8$	31.58 ± 0.67	27.97 ± 0.94
$F = 16$	$\mathbf{32.43 \pm 1.12}$	$\mathbf{28.78 \pm 0.83}$

Table 7: Scratch-training exact-match accuracy (%). Values are mean $\pm$ sample standard deviation over seeds 43–45. Dropout ($p = 0.025$) and stabilized OAT ($\lambda = 0.03$) are selected only on `arithmetic__mixed` with seed 42 and then frozen. Bold indicates the highest mean in each row.

Task	NTP	Dropout	Stabilized OAT
Mixed	30.61 ± 3.81	31.30 ± 2.16	$\mathbf{31.78 \pm 1.13}$
Linear 1D	75.47 ± 4.52	76.24 ± 2.72	$\mathbf{77.82 \pm 6.68}$
Add/Sub	82.91 ± 7.18	86.65 ± 3.48	$\mathbf{86.79 \pm 2.49}$

learning rate and a source-conditioned tracking term that encourages the target frame selected at position u to remain preferred at $u + 1$. One possible explanation is that local continuity across adjacent token positions in pre-trained representations helps resist these solutions during fine-tuning. Tracking supplies an explicit local consistency constraint when representations are learned from scratch. Appendix C gives the objective and stabilization diagnostics.

We train a 12-layer decoder-only Transformer with hidden size 256 and 8 attention heads on three modules of the DeepMind Mathematics dataset (Saxton et al., 2019): `arithmetic__mixed`, `algebra__linear_1d`, and `arithmetic__add_sub_multiple`. All runs use AdamW (Loshchilov & Hutter, 2019) with weight decay 0.1. We compare stabilized OAT with residual dropout as a generic additional regularizer rather than with STP, whose published evaluation is on fine-tuning pre-trained language models (Huang et al., 2026b); transferring STP to this randomly initialized character-level setting would introduce an additional scratch-specific recipe.

Residual dropout and stabilized OAT are tuned independently only on `arithmetic__mixed` using seed 42, selecting $p = 0.025$ and $\lambda = 0.03$, and are then frozen. The larger dropout rates $p = 0.05$ and 0.1 both reduce accuracy in this selection setting. Table 7 reports disjoint seeds 43–45. Linear 1D and Add/Sub therefore test transfer without retuning.

Stabilized OAT obtains the highest mean on all three tasks, improving over NTP by 1.16, 2.35, and 3.89 percentage points. Its margins over independently tuned dropout are modest, and it does not outperform dropout on every seed. These results provide descriptive evidence of utility beyond pre-trained fine-tuning. Removing tracking again makes the auxiliary objective easy to satisfy, but eliminates the downstream improvement (Appendix C).

4 Related Work

Future-aware language modeling. Language-model training objectives increasingly supplement next-token prediction with targets that reach beyond the immediate next token. Multi-token prediction trains a shared Transformer trunk to predict several future tokens through parallel output heads (Gloeckle et al., 2024), while Future Summary Prediction predicts compact summaries of longer-range future context (Mahajan et al., 2026). Other approaches seek to shape the model's internal representations. Belief State Transformers combine forward and backward predictive objectives to encourage compact states that support goal-directed generation (Hu et al., 2025); NextLat predicts the next latent state conditioned on the next token (Teoh et al., 2026); and Semantic Tube Prediction imposes a geometric prior on hidden-state trajectories (Huang et al., 2026b). Complementary diagnostic work uses the Prediction of Hidden States (PHi) bottleneck to measure future hidden-state

predictability (Herrmann et al., 2025). OAT uses future hidden representations as targets, with correspondences selected through off-diagonal matching across learned projections.

Predictive representation learning. Predictive representation learning has a broader history outside language-model fine-tuning. Contrastive Predictive Coding learns representations from predictive relationships in latent space (van den Oord et al., 2018), while data2vec predicts contextualized latent targets from masked inputs across speech, vision, and language (Baevski et al., 2022). Joint-Embedding Predictive Architectures similarly predict target representations from partial context in images (Assran et al., 2023) and video (Bardes et al., 2024). LLM-JEPA applies joint-embedding prediction to language models using paired inputs such as natural-language descriptions and code (Huang et al., 2026a). OAT instead constructs its views through multiple learned readouts of each hidden state. A complementary line of work exploits temporal structure directly: Slow Feature Analysis extracts decorrelated features that vary slowly in time (Wiskott & Sejnowski, 2002), and Dynamical Components Analysis identifies low-dimensional projections that preserve predictive information between the past and future of a time series (Clark et al., 2019).

Redundancy reduction. Agreement objectives in self-supervised learning are commonly paired with mechanisms designed to discourage uninformative representations. Barlow Twins combines agreement with redundancy reduction through cross-correlation structure (Zbontar et al., 2021), while VICReg uses invariance, variance, and covariance terms (Bardes et al., 2022). OAT measures positive directional overlap among multiple views of the same token and couples it to off-diagonal temporal matching. This local construction uses neither negative examples nor batch-level covariance statistics. An audit of representation objectives on NL-RX also finds that altered hidden-state geometry need not yield reliable exact-match gains (Sengupta, 2026).

Transformer representations. Next-token training can produce structured internal states whose geometry reflects belief states (Shai et al., 2024). Models trained on the same task can also learn distinct state-tracking mechanisms (Li et al., 2025). Other analyses show that when a useful representation becomes available can matter to subsequent computation (Lepori et al., 2025), and recent work studies how abstract features that appear unnecessary for the immediate next token can nevertheless emerge under the next-token objective (Rofin et al., 2026). The geometric interpretation of concept representations depends on the choice of inner product (Park et al., 2024). OAT does not identify its frame views with semantic factors; it regularizes their within-state and temporal relationships while leaving the autoregressive inference graph unchanged.

5 LIMITATIONS

Our evaluation covers instruction-tuned models between 0.5B and 1B parameters, with one full-data Llama-3.2-3B comparison, on NL-RX generation and mathematical reasoning. Gains vary with data fraction, and the 3B results show only a small positive mean gain on GSM8K and essentially no gain on GSM1K. These experiments do not establish broad scaling behavior or uniform benefits across task domains.

The experiments do not identify what information is matched across OAT views. The ablations support both structural constraints, and the fixed-routing diagnostic shows that one simple position-conditioned rule does not fully explain the matches. Neither establishes semantic correspondence or the mechanism of improvement. Future targets are computed under teacher forcing and contain information unavailable at the source position. The gains may therefore reflect preservation of task-relevant state, supervision from future-conditioned representations, or an interaction of these effects.

Finally, the training-from-scratch experiments are deliberately controlled and should not be interpreted as evidence that OAT directly improves large-scale language-model pretraining. They use a small character-level Transformer on selected DeepMind Mathematics tasks and require a scratch-specific tracking stabilization. Extending OAT to large-scale pretraining, and understanding when such stabilization is necessary, remain open directions.

6 CONCLUSION

We introduced Off-Diagonal Atlas Transport, a training-time regularizer that shapes temporal relationships through multiple learned projections of hidden states. OAT combines off-diagonal matching with within-state redundancy suppression, allowing correspondences across different projection indices without requiring the hidden-state trajectory itself to be locally straight. The auxiliary projections are discarded after training, leaving the autoregressive inference graph unchanged and adding no inference-time computation.

In low-data fine-tuning across three model families and three tasks, OAT achieved higher mean accuracy than NTP in all nine combinations. While STP remained stronger on regular-expression generation, OAT showed its clearest advantage over STP on mathematical reasoning, suggesting that different tasks may benefit from different temporal inductive biases.

The structural ablations reveal two distinct ways to satisfy temporal matching without retaining its downstream benefit. Allowing same-index matches permits views to rely on persistent frame identity, whereas removing redundancy suppression allows different views to become highly aligned. In both cases, the auxiliary loss approaches zero, yet downstream gains largely disappear. These findings highlight the joint design of allowable correspondences and within-state constraints as an important consideration for temporal representation objectives.

In controlled experiments from random initialization, stabilized OAT also improves mean accuracy over NTP on three mathematics tasks. The need for additional tracking in this setting raises a broader question: how should flexibility in cross-view correspondence be balanced with temporal consistency as representations are learned?

This question extends beyond language modeling. A natural next step is to explore how cross-view matching, redundancy control, and temporal consistency should be combined in settings such as video, sensor time series, and sequential decision-making. Such studies could examine how suitable constraints vary with data dynamics and the representations available at initialization. More broadly, OAT motivates predictive representation objectives that accommodate changing correspondences while constraining shortcuts that make auxiliary agreement uninformative.

AI USE STATEMENT

Large language models were used to assist with manuscript drafting and language polishing, literature retrieval and discovery, and aspects of research execution. In particular, they were used to help refine the presentation of the method, review experimental and ablation plans, assist with interpreting and calibrating empirical claims, organize related work, and check consistency between the manuscript and implementation. The research ideas and methodological choices originated with the authors. All experimental results, citations, technical claims, and final methodological and editorial decisions were independently reviewed and verified by the authors. The authors take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

Training settings and coefficient-selection procedures are detailed in Section 3.1 and Appendices A and C. The implementation used in this work is included as anonymized supplementary material. We intend to release a public repository with the camera-ready version.

REFERENCES

Allen Institute for AI. OLMo-2-0425-1B-Instruct model card. Hugging Face model card, 2025. URL https://huggingface.co/allenai/OLMo-2-0425-1B-Instruct.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, pp. 15619–15629, 2023.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In *ICML*, pp. 1298–1312, 2022.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

David G. Clark, Jesse A. Livezey, and Kristofer E. Bouchard. Unsupervised discovery of temporal structure in noisy data with dynamical components analysis. In *NeurIPS*, 2019.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In *ICML*, pp. 15706–15734, 2024.

Vincent Herrmann, Róbert Csordás, and Jürgen Schmidhuber. Measuring in-context computation complexity via hidden state prediction. In *ICML*, pp. 23063–23082, 2025.

Edward S. Hu, Kwangjun Ahn, Qinghua Liu, Haoran Xu, Manan Tomar, Ada Langford, Dinesh Jayaraman, Alex Lamb, and John Langford. The Belief State Transformer. In *ICLR*, 2025.

Hai Huang, Yann LeCun, and Randall Balestriero. LLM-JEPA: Large language models meet joint embedding predictive architectures. In *ICLR*, 2026a.

Hai Huang, Yann LeCun, and Randall Balestriero. Semantic Tube Prediction: Beating LLM data efficiency with JEPA. *arXiv preprint arXiv:2602.22617*, 2026b.

Michael A. Lepori, Michael Curtis Mozer, and Asma Ghandeharioun. Racing thoughts: Explaining contextualization errors in large language models. In *NAACL*, pp. 3020–3036, 2025.

Belinda Z. Li, Zifan Carl Guo, and Jacob Andreas. (How) do language models track state? In *ICML*, pp. 34429–34452, 2025.

Nicholas Locascio, Karthik Narasimhan, Eduardo DeLeon, Nate Kushman, and Regina Barzilay. Neural generation of regular expressions from natural language with minimal domain knowledge. In *EMNLP*, pp. 1918–1923, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Divyat Mahajan, Sachin Goyal, Badr Youbi Idrissi, Mohammad Pezeshki, Ioannis Mitliagkas, David Lopez-Paz, and Kartik Ahuja. Beyond multi-token prediction: Pretraining LLMs with future summaries. In *ICLR*, 2026.

Meta. Llama 3.2 model card. Model card, 2024. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *ICML*, pp. 39643–39666, 2024.

Mark Rofin, Jalal Naghiyev, and Michael Hahn. Understanding the emergence of seemingly useless features in next-token predictors. In *ICLR*, 2026.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. In *ICLR*, 2019.

Biswa Sengupta. Representation without reward: A JEPA audit for LLM fine-tuning. *arXiv preprint arXiv:2605.15394*, 2026.

Adam S. Shai, Sarah E. Marzen, Lucas Teixeira, Alexander Gietelink Oldenziel, and Paul M. Riechers. Transformers represent belief state geometry in their residual stream. In *NeurIPS*, 2024.

Jayden Teoh, Manan Tomar, Kwangjun Ahn, Edward S. Hu, Tim Pearce, Pratyusha Sharma, Akshay Krishnamurthy, Riashat Islam, Alex Lamb, and John Langford. Next-latent prediction transformers learn compact world models. *arXiv preprint arXiv:2511.05963v4*, 2026.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Laurenz Wiskott and Terrence J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115v2*, 2024.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-supervised learning via redundancy reduction. In *ICML*, pp. 12310–12320, 2021.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Charlotte Zhuang, Dylan Slack, Qin Lyu, Sean Hendryx, Russell Kaplan, Michele Lunati, and Summer Yue. A careful examination of large language model performance on grade school arithmetic. In *NeurIPS*, pp. 46819–46836, 2024.

A Additional Fine-Tuning Details

To make coefficient selection comparable across OAT and STP, we tune both only on Llama at $1/16$ data using the same five matched seeds. For each objective and task family, we sweep the auxiliary coefficient on a factor-of-two grid starting at 0.01 ($0.01, 0.02, 0.04, \ldots$) and select the coefficient with the highest five-seed mean test accuracy. SYNTH is used to select the RX coefficient and GSM8K to select the math coefficient; TURK inherits the RX choice and GSM1K inherits the coefficient of its GSM8K-trained checkpoint. This protocol selects $\lambda_{\mathrm{OAT}} = 0.02$ for RX and 0.04 for math, and independently selects $\lambda_{\mathrm{STP}} = 0.32$ for RX and 0.02 for math. The selected coefficients are then transferred unchanged to OLMo, Qwen, and the Llama-3.2-3B scale check.

Table 8: OAT coefficient sweep on Llama at $1/16$ data under the common factor-of-two selection protocol. Values are accuracy in percent, mean $\pm$ sample standard deviation over five seeds. The selected coefficients are bold.

λ_{OAT}	SYNTH	GSM8K
0.01	67.31 ± 0.78	30.46 ± 1.44
0.02	$\mathbf{67.76 \pm 0.66}$	32.07 ± 1.17
0.04	67.75 ± 0.68	$\mathbf{32.43 \pm 1.12}$
0.08	67.30 ± 0.28	31.27 ± 0.59

For the no-end-of-turn SYNTH control discussed in Section 3.1, STP reaches 52.50 ± 19.41 at $\lambda = 0.02$ and 11.48 ± 2.12 at $\lambda = 0.32$, compared with 29.51 ± 1.82 for NTP without end-of-turn supervision and 66.82 ± 0.44 for the end-of-turn-supervised NTP baseline. Frequent correct-prefix-but-overlong generations show that termination behavior dominates the exact-match difference, motivating the common EOT-supervised protocol used in the main comparison.

B Fixed-Routing Diagnostic

The structural ablations highlight two simple shortcuts that OAT's constraints are designed to suppress, but its off-diagonal matching could still reflect a content-independent routing rule. In particular, the selected target frame might be largely determined by the sampled layer ℓ, the absolute source and target positions (t, u), and the source-frame identity f, rather than by example-specific representations. We test this possibility on Llama fine-tuned on GSM8K with the training minimum gap $\delta = 10$.

We split the GSM8K test examples deterministically into equal calibration and evaluation halves. The diagnostic uses a predefined grid spanning multiple absolute source positions and temporal offsets, fixed independently of the training sampler; we retain only source–target position pairs with at least 50 valid content-token observations in each half. For each retained (ℓ, t, u, f), let

$$g_i^*(\ell, t, u, f) = \arg\max_{g \neq f} C_{fg}^{(i,\ell)}(t, u) \tag{5}$$

be the adaptive target-frame choice on calibration example i. We define the corresponding fixed route as its modal choice,

$$\hat{\pi}_{\ell,t,u}(f) = \mathrm{mode}_{i \in \mathcal{D}_{\mathrm{cal}}}\, g_i^*(\ell, t, u, f). \tag{6}$$

This gives a deterministic lookup table that is allowed to depend on layer, absolute source and target positions, and source-frame identity, but not on the held-out example.

On the evaluation half, we measure the fraction of adaptive target-frame choices reproduced by this table,

$$A_{\mathrm{fixed}} = \frac{1}{F} \sum_f \mathbf{1}\left[\arg\max_{g \neq f} C_{fg} = \hat{\pi}_{\ell,t,u}(f)\right], \tag{7}$$

and compare the ordinary adaptive score

$$P_{\max} = \frac{1}{F} \sum_f \max_{g \neq f} C_{fg} \tag{8}$$

with the score obtained by following the same fixed route,

$$P_{\text{fixed}} = \frac{1}{F} \sum_{f} C_{f,\hat{\pi}_{\ell,t,u}(f)}. \tag{9}$$

We first average over held-out observations for each retained position pair, then average position pairs equally and finally average across the sampled OAT layers. Table 9 reports the resulting seed-level means.

Table 9: Fixed-routing diagnostic on Llama/GSM8K with $\delta = 10$. Values are mean $\pm$ sample standard deviation over five seeds. The fixed route is the modal adaptive target-frame choice estimated on the calibration half and is evaluated on held-out examples.

Diagnostic	Value
Fixed-route agreement A_{fixed}	0.373 ± 0.046
Adaptive score $P_{\max}$	0.152 ± 0.013
Fixed-route score P_{fixed}	0.047 ± 0.015

The position-conditioned route captures a substantial but incomplete fraction of held-out frame choices, and its temporal score remains well below the adaptive score. Thus OAT exhibits fixed positional routing tendencies, but this restricted deterministic map does not account for the full matching behavior. The diagnostic is deliberately narrow: it does not identify semantic correspondences, establish that routing is purely content-dependent, or rule out more complex content-independent mechanisms involving variables beyond layer, frame identity, and absolute positions.

C Scratch-Training Details and Stabilization

Training protocol. We train a 12-layer decoder-only Transformer with hidden size 256 and 8 attention heads on character-level sequences from three modules of the DeepMind Mathematics dataset: `arithmetic__mixed`, `algebra__linear_1d`, and `arithmetic__add_sub_multiple`. Each run uses 300,000 training examples, batch size 64, and 200,000 optimization steps. We use AdamW with main-model learning rate 2×10^{-4}, 2,000 warmup steps followed by cosine decay to 0.1 of the peak rate, weight decay 0.1, $\beta = (0.9, 0.98)$, gradient clipping at norm 1, and no label smoothing. For OAT we use $F = 8$ frames of dimension 64, sample from layers 4–7, use up to 24 source positions and $K = 2$ future targets per source, set $\tau = 0.1$, and impose no minimum source–target gap ($\delta = 0$). The projection frames use learning rate 10^{-5} to reduce their ability to rapidly absorb the auxiliary objective through a projector-only shortcut.

Scratch training additionally uses a source-conditioned tracking term. For a source-frame index f and sampled target u, let

$$g^*(t, u, f) = \arg\max_{g \neq f} C_{fg}(t, u) \tag{10}$$

be the off-diagonal target-frame index chosen by OAT. When $u + 1$ is a valid content position, we encourage the same target-frame identity to remain preferred one step later:

$$\mathcal{L}_{\text{track}} = \mathbb{E}_{t,u,f} \left[\max_{g \neq g^*(t,u,f)} C_{fg}(t, u+1) - C_{f,g^*(t,u,f)}(t, u+1) \right]_+ . \tag{11}$$

The target representations and the discrete choice g^* are stop-gradient, as in the OAT temporal term. The scratch objective is

$$\mathcal{L}_{\text{scratch}} = \mathcal{L}_{\text{NTP}} + \lambda(t) \left(\mathcal{L}_{\text{OAT}} + w_{\text{track}} \mathcal{L}_{\text{track}} \right), \tag{12}$$

with $w_{\text{track}} = 10$. The common coefficient $\lambda(t)$ is ramped from zero to its target value between 5% and 25% of training, matching the fine-tuning schedule.

We use `arithmetic__mixed` as the sole hyperparameter-selection task. On seed 42, we independently sweep residual dropout over $p \in \{0.0125, 0.025, 0.05, 0.1\}$ and the stabilized OAT coefficient over $\lambda \in \{0.01, 0.03, 0.1\}$, selecting by exact-match accuracy at the final 200,000-step checkpoint. This selects $p = 0.025$ and $\lambda = 0.03$; the stronger dropout settings $p = 0.05$ and 0.1 reduce selection-task accuracy. We then freeze both strengths and report disjoint seeds 43–45 in Table 7.

Tracking ablation. We isolate the role of tracking on `arithmetic__mixed` by removing $\mathcal{L}_{\text{track}}$ while leaving the scratch architecture, projector learning rate, sampling rule, and $\lambda = 0.03$ unchanged. Table 10 reports downstream accuracy together with diagnostics averaged over the final logging interval.

Table 10: Tracking ablation on scratch-trained Mixed. Accuracy is mean $\pm$ sample standard deviation over seeds 43–45; geometry columns are late-training means. Occupancy is the maximum target-frame occupancy, whose minimum is $1/F = 0.125$ for $F = 8$.

Configuration	Accuracy	$\mathcal{L}_{\text{OAT}}$	$P_{\text{off-diag}}$	Occupancy	$\mathcal{L}_{\text{track}}$
No tracking	30.56 ± 3.84	0.0019	0.714	0.146	0.634
+ Tracking	$\mathbf{31.78 \pm 1.13}$	0.165	0.188	0.320	0.107

Without tracking, OAT drives its own loss almost to zero while leaving downstream accuracy essentially unchanged from NTP (30.56 versus 30.61). The resulting geometry is highly structured rather than collapsed: within-state redundancy remains low, the off-diagonal temporal score rises to 0.714, and target-frame occupancy falls to 0.146, close to its minimum of $1/F = 0.125$. At the same time, the large tracking violation shows that the target-frame identity selected at u is generally not preserved at $u + 1$. This pattern is consistent with a rapidly changing routing-like solution that makes off-diagonal matching easy, without by itself identifying the variables governing that routing.

With tracking, the retained OAT loss is higher, the off-diagonal score and occupancy move away from the near-permutation regime, and the tracking violation decreases substantially. The downstream gain from tracking is modest and seed-variable on its own, but the contrast mirrors the fine-tuning ablations: making the auxiliary geometry easier to optimize does not imply better task performance.