Supplementary Materials: Supplementary Materials for "Equivariant Transition Matrices for Explainable Deep Learning: A Lie Group Linearization Approach"

Pavlo Radiuk, Oleksander Barmak, Leonid Bedratyuk and Iurii Krak

S1. Introduction

The purpose of this supplement is twofold. First, it records the concrete implementation choices that connect the mathematical formulation in the main manuscript to the released software project, so that readers can verify which quantities are measured, configured, and logged rather than inferred. Second, it collects analyses that are too detailed or too tangential for the main narrative but are directly responsive to reviewer concerns, including generator-estimation bias, solver diagnostics, multi-generator extensions, and discussion of domains beyond MNIST. Throughout, we distinguish carefully between values extracted from the archived repository and compact controlled analyses added during revision.

This Supplementary Material (Section S1) documents implementation details, reproducibility metadata, and additional analyses requested during revision. The revisions are grounded in the released project archive: the training configuration is taken from `configs/mnist.yaml`, the synthetic setup is taken from `configs/synthetic.yaml`, and the reported solver statistics are taken from the archived run and metrics manifests under `outputs/`. The present supplement therefore distinguishes explicitly between quantities reproduced from the supplied software project and compact controlled analyses added during revision to clarify generalization beyond the main $SO(2)$ experiments.

S2. Detailed Implementation Specifications

Section S2 ties the revised manuscript directly to the supplied project archive. Wherever possible, the descriptions below mirror the released configuration files and code paths rather than idealized pseudo-configurations. This was an explicit revision goal because reviewer comments correctly requested that the computational workflow be reproducible from the repository itself, not reconstructed from memory or from a schematic description.

S2.1. Released configuration files and deterministic settings

The repository exposes the full experimental configuration through plain-text YAML files and stores numerical outputs as JSON artifacts. This design makes the workflow auditable without hidden notebook state. Table S1 summarizes the configuration values used in the archived MNIST experiment that underlies the revised manuscript.

S2.2. CNN architecture for MNIST feature extraction

The archived formal model differs from the draft description that accompanied the earlier submission. The released code in `src/mnist/model.py` implements the two-convolution architecture summarized in Table S2. We report the repository version here to ensure exact reproducibility.

The archived training history stored in outputs/mnist/models/training_history.json shows that this model reaches 99.11% test accuracy after five epochs. ETM is then fitted on the frozen penultimate representation, so the explanatory stage remains strictly post hoc.

Table S1. Released MNIST configuration extracted from `configs/mnist.yaml`.

Item	Value
Random seed	42
Training batch size	256
Generator-estimation batch size	128
Feature dimension k	490
Training epochs	5
Optimizer	Adam
Learning rate	10^{-3}
Weight decay	0
Generator truncation threshold τ	10^{-8}
Transition regularization weight λ	0.5 in the reference run; sweep grid $\{0, 10^{-3}, 10^{-2}, 10^{-1}, 0.5, 1, 5, 10\}$
LSQR settings	iteration limit 200, absolute tolerance 10^{-6}, relative tolerance 10^{-6}
Evaluation range	rotations from -90° to 90° in 10° increments
Execution device in archived run	CPU

Table S2. CNN architecture used for the released MNIST experiment.

Layer	Configuration	Output shape
Input	normalized grayscale image	$1 \times 28 \times 28$
Conv1 + ReLU	32 filters, 3×3, padding 1	$32 \times 28 \times 28$
MaxPool	2×2	$32 \times 14 \times 14$
Conv2 + ReLU	64 filters, 3×3, padding 1	$64 \times 14 \times 14$
MaxPool	2×2	$64 \times 7 \times 7$
Dropout	$p = 0.25$	$64 \times 7 \times 7$
FC1 + ReLU	$64 \cdot 7 \cdot 7 \to 490$	490
Dropout	$p = 0.25$	490
FC2	$490 \to 10$	10 logits

S2.3. Rotation pipeline and generator estimation in the released code

For MNIST, the released implementation performs rotations with PyTorch's affine_grid / grid_sample operator in bilinear mode with zero padding. The generator matrices are then estimated by central finite differences with step $\varepsilon = 10^{-4}$ using the routines `compute_A_and_dA` and `compute_B_and_dB` in `src/mnist/generators.py`. Thus, the revision clarifies an important point: the released code uses a differentiable rotation operator but evaluates the derivative numerically by symmetric finite differences rather than by an exact Jacobian-vector product.

For the synthetic benchmark, the repository follows the bridge construction described in the main text. Each high-dimensional feature cloud is embedded into $\mathbb{R}^2$ by metric MDS, rotated by a small angle $\varepsilon = 10^{-2}$, and mapped back to the original feature space by an affine linear decoder. The resulting finite differences are then regressed onto the original features to estimate J^A and J^B.

S2.4. Clarifications on temporal alignment, collinearity, and uncertainty

Some reviewer comments were phrased in the language of heterogeneous environmental predictors. That situation does not arise in the present study. For each datum, the FM and MM representations are computed from the same sample and from the same preprocessing pipeline, so there is no cross-year or cross-period alignment issue analogous to mixing temporally mismatched satellite and climate layers.

The closest analogue to predictor multicollinearity is redundancy in the learned feature coordinates. This redundancy is handled in two places. First, the baseline transition matrix uses a pseudoinverse/SVD solution rather than naive normal-equation inversion. Second, ETM adds structured regularization through the commutator penalty, which discourages unstable directions that fit noise but violate the assumed symmetry.

The revised manuscript reports empirical uncertainty in the form of standard deviations, confidence intervals for mean SSIM/PSNR, and bootstrap variability for the synthetic benchmark. We emphasize that these quantities describe variability of evaluation metrics, not a posterior distribution over the transition matrix itself. A full uncertainty-aware ETM, for example by bootstrap ensembles or Bayesian linear operators, remains an important extension rather than a capability claimed in the present paper.

S3. Extended Computational Analysis

The main paper summarizes computational complexity at a high level, but practical scalability depends on a more concrete decomposition of stored arrays, operator evaluations, and solver diagnostics. Section S3 therefore records the memory-bearing objects and the archived runtime signals separately. This distinction matters because the ETM system is large in equation count but modest in explicitly stored state once the Kronecker structure is handled implicitly. Table S3 decomposes the major stored arrays, while Table S4 reports the archived solver behavior.

S3.1. Storage requirements for the MNIST-scale problem

The most important scalability fact is that ETM never materializes the full Kronecker matrix in the released MNIST implementation. Table S3 therefore reports the storage cost of the explicitly stored arrays only.

Table S3. Analytical storage cost of the explicitly stored MNIST arrays in double precision. Decimal megabytes are reported for consistency with the run logs.

Component	Size (MB)	Comment
Feature matrix $A \in \mathbb{R}^{60000\times490}$	235.2	frozen FM features
Feature matrix $B \in \mathbb{R}^{60000\times784}$	376.3	pixel-space MM features
Generator $J^A \in \mathbb{R}^{490\times490}$	1.92	FM infinitesimal action
Generator $J^B \in \mathbb{R}^{784\times784}$	4.92	MM infinitesimal action
Vectorized solution $u \in \mathbb{R}^{384160}$	3.07	$\mathrm{vec}(T)$

S3.2. Runtime and solver diagnostics from the archived run

Table S4 collects the project-level computational diagnostics extracted from the archived stage-2 MNIST run. The numbers correspond to the released configuration rather than to a hypothetical high-end workstation benchmark.

S4. Computational Complexity and Scalability Analysis

The complexity discussion in Section S4 should be read as a model of resource scaling, not a claim about one hardware setup only. The released run provides one empirical point,

Table S4. Runtime and solver diagnostics for the archived MNIST ETM run.

Metric	Value
Problem dimensions (m, k, ℓ)	$(60000, 490, 784)$
Number of unknowns $k\ell$	384,160
Number of linear equations $m\ell + k\ell$	47,424,160
LSQR iterations at $\lambda = 0.5$	127
Residual norm r_1	905.09
Estimated condition number $\hat{\kappa}$	2.46×10^3
Reported ETM solve time	47.3 s
Inference-time explanation cost after fitting	one FM forward pass plus one matrix multiplication

while Equation (1) and Equation (2) describe how the cost moves with m, k, ℓ, and the number of generators. This separation between empirical diagnostics and analytical scaling is useful when readers want to extrapolate from MNIST to larger representation layers.

Let $n = k\ell$ denote the number of unknown transition parameters. If the system matrix were formed explicitly, the fidelity block alone would contain $m\ell$ rows and n columns, which is infeasible for modern representation sizes. The implicit operator formulation avoids this bottleneck because the Kronecker identities in Equation (1) permit matrix–vector products to be evaluated without explicit materialization:

$$(A \otimes I_\ell)\,\mathrm{vec}(T) = \mathrm{vec}(TA^\top), \qquad \Big((J_i^A)^\top \otimes I_\ell - I_k \otimes J_i^B\Big)\mathrm{vec}(T) = \mathrm{vec}(TJ_i^A - J_i^B T). \quad (1)$$

Using Equation (1), each matrix-vector product therefore costs, as stated in Equation (2),

$$O\Big(mk\ell + r(k^2\ell + k\ell^2)\Big), \quad (2)$$

rather than the cost of multiplying by a dense $(m\ell + rk\ell) \times k\ell$ matrix.

This complexity statement extends directly to multiple generators. Increasing from one generator ($SO(2)$) to r generators scales the symmetry block linearly in r. The practical limit is thus not the group itself but the size and structure of the generator matrices in the chosen feature spaces. When the generators are sparse, low rank, or have fast transforms, the operator can be accelerated further. When they are dense and very high dimensional, block-iterative methods, sketching, or concept-space MMs become attractive.

S5. Generator Estimation with Regularization

Generator estimation is the methodological hinge of ETM. If the infinitesimal action is estimated poorly, the transition operator may be regularized toward the wrong structure. Section S5 therefore expands this stage beyond the terse description in the main text by reporting explicit regularized objectives, proxy-space diagnostics, and small controlled experiments in which the true generators are known.

The generator estimation step solves the regularized problems in Equation (3) before ETM fitting:

$$J_i^A = \arg\min_J \Big\|\Delta A_i - AJ^\top\Big\|_F^2 + \alpha_J \|J\|_F^2, \qquad J_i^B = \arg\min_J \Big\|\Delta B_i - BJ^\top\Big\|_F^2 + \alpha_J \|J\|_F^2. \quad (3)$$

In the released MNIST pipeline, this regression is implemented in reduced form through SVD-truncated pseudoinverses of $A^\top A$ and $B^\top B$ with threshold $\tau = 10^{-8}$.

S5.1. Proxy-space bias in the synthetic MDS bridge

The synthetic protocol deliberately introduces a proxy space because the abstract matrices A and B do not come with an intrinsic object-level action. This makes the MDS bridge methodologically useful but also susceptible to distortion. To quantify this effect, we computed four diagnostics directly from the archived synthetic outputs: pairwise-distance distortion of the MDS embedding, affine decoder reconstruction error, and the normalized generator-fit residuals of Equation (4)

$$\eta_A = \frac{\|\Delta A - A(J^A)^\top\|_F}{\|\Delta A\|_F}, \qquad \eta_B = \frac{\|\Delta B - B(J^B)^\top\|_F}{\|\Delta B\|_F}. \tag{4}$$

Table S5. Proxy-space diagnostics for the synthetic MDS bridge, computed from the archived project outputs. Lower values indicate less distortion.

Diagnostic	**FM space** A	**MM space** B
Pairwise-distance distortion after MDS	0.0187	0.0134
Affine decoder reconstruction residual	0.0882	0.0913
Normalized generator-fit residual η	0.1786	0.2256

Table S5 supports two conclusions. First, the embedding step is not lossless and therefore should not be interpreted as an oracle symmetry model. Second, the distortion is moderate rather than catastrophic, which explains why the synthetic experiment remains informative: the proxy rotation induces a stable local tangent direction even though it is only approximate. The revised manuscript now makes this approximation explicit and treats the synthetic benchmark as a controlled proxy study rather than as a literal object-space group action.

S6. Synthetic Experiment with Known Group Action

This controlled benchmark was added to address the concern that the main synthetic study relies on an MDS proxy. By constructing FM and MM spaces with analytically known $SO(2)$ actions, we can test ETM in a setting where the symmetry is exact by design. The exercise is intentionally small, but it isolates the core effect of the commutator penalty without confounding geometric projection error.

To complement the MDS-proxy synthetic benchmark, we added a small controlled $SO(2)$ experiment in which the generators are known analytically. Two-dimensional latent variables were mapped into FM and MM spaces by random invertible linear transforms, additive Gaussian noise was injected, and the true intertwiner T_{true} was therefore known exactly. Table S6 reports averages over 200 random trials with noise level $\sigma = 0.1$ and $\lambda = 0.5$.

Table S6. Controlled $SO(2)$ benchmark with known generators. Reported values are means over 200 random trials.

Method	**MSE**	**SymDef**	**Relative** T **error**
Baseline least squares	0.0984	0.7700	0.1026
ETM with known $SO(2)$ generator	0.1013	0.1882	0.0905

This experiment is useful because it removes the proxy-space ambiguity altogether. The large SymDef reduction with only a small MSE increase confirms that the gain observed in the main paper is not an artifact of the MDS construction.

S7. Multi-Generator Extension Analysis

The $SE(2)$ extension reported in Section S7 is not presented as a definitive benchmark but as a structural sanity check on multi-generator scaling. Its value lies in demonstrating that the ETM objective and diagnostics extend mechanically from one generator to several, while also revealing where the practical burden increases: the estimation and validation of multiple generators, not the algebra of the penalty itself.

S7.1. $SE(2)$ extension in a controlled linear benchmark

To address the question of generality beyond a one-parameter group, we constructed a compact $SE(2)$ benchmark using the standard 3×3 homogeneous-coordinate generators for translation in x, translation in y, and in-plane rotation. Random similarity transforms were then used to induce corresponding FM and MM representations, and ETM was fit either with only the rotational generator or with all three generators. Table S7 reports averages over 100 random trials with additive noise $\sigma = 0.1$.

Table S7. Controlled $SE(2)$ benchmark with three generators. SymDef is aggregated across all three commutator terms.

Method	MSE	SymDef	Relative T error
Baseline least squares	0.0976	1.9028	0.1315
ETM with $SO(2)$ generator only	0.1018	1.4233	0.1225
ETM with full $SE(2)$ generator set	0.1072	0.5066	0.1124

The trend is consistent with the theory: adding generators strengthens structural control and improves recovery of the true intertwiner, but it also narrows the feasible set and therefore increases MSE slightly. This controlled experiment does not replace a full large-scale $SE(2)$ benchmark, but it provides direct evidence that the formulation scales naturally from one generator to several.

S7.2. Implications for 3D point clouds

The same principle should carry over to point clouds and other irregular geometric data, provided that both the FM and MM representations admit compatible actions of the relevant 3D transformation group. In that setting, ETM would not replace architecture-level geometric inductive biases; instead, it would act as a post-hoc bridge on top of learned latent features. This is especially relevant when the backbone is already specialized for point-cloud understanding, such as transformer and physics-informed models that fit local geometric structure [1,2]. The main practical difficulty is not the ETM objective itself, but the construction of MM features and generators that remain meaningful under SE(3)-style transformations. We therefore view 3D point clouds as a promising but still application-specific direction.

S8. Comprehensive Hyperparameter Studies

The reviewer requested a more principled account of λ selection than simple validation-set performance. Accordingly, Section S8 now includes both the released sweep values and an unsupervised operating-point heuristic based on the discrete L-curve. The two views are complementary: the sweep reveals the empirical Pareto frontier in the archived run, while the knee criterion provides a deployment rule when labeled validation data are unavailable.

S8.1. Hyperparameter policy for the released workflow

The revised manuscript now distinguishes clearly between backbone training hyperparameters and ETM-specific hyperparameters. The released workflow does not perform large-scale automated hyperparameter optimization of the CNN with grid search, randomized search, or Bayesian optimization. Instead, the backbone is trained once with the fixed configuration in Table S1. The explicit tuning effort is focused on the single ETM regularization parameter λ, for which the project ships the sweep grid used in the paper.

S8.2. Released λ sweep for MNIST

Table S8 reports the archived MNIST λ sweep. It confirms the qualitative trade-off discussed in the main text: equivariance improves monotonically, while fidelity changes only gradually until the largest regularization values.

Table S8. Released MNIST λ sweep.

λ	SymDef	LSQR iters	Comment
0	141.19	–	baseline transition matrix
0.001	84.27	100	substantial symmetry improvement at almost unchanged fidelity
0.01	81.67	100	similar regime to $\lambda = 10^{-3}$
0.1	64.79	100	clear move toward the structural regime
0.5	38.65	127	operating point used in the main experiments
1	30.38	100	further improvement
5	22.75	100	diminishing returns relative to fidelity loss
10	21.05	100	strongest regularization in released sweep

S9. Sensitivity Analysis for Generator Estimation

Finite-difference sensitivity is reported separately because it interacts with two different failure modes: numerical cancellation for very small steps and breakdown of local linearization for very large ones. The revised supplement makes this trade-off explicit so that readers can treat ε as a controlled numerical parameter rather than as an opaque implementation constant.

The released code fixes the central-difference step to $\varepsilon = 10^{-4}$ for MNIST and to $\varepsilon = 10^{-2}$ for the synthetic MDS proxy benchmark. Because the repository does not ship a dedicated step-size sweep, we do not claim a global optimum for ε. Instead, the revision adds two safeguards. First, the main manuscript now states explicitly that generator estimation should be validated by residual diagnostics such as η_A and η_B. Second, the controlled known-generator benchmark above shows that ETM remains beneficial when the generators are exact, which isolates the value of the commutator constraint from any particular finite-difference choice.

S10. Computational Performance and Scalability

The runtime table in Section S10 is intentionally pragmatic. It records not only the total ETM solve time but also the scale of the linear system and the inference-time footprint once the operator has been fitted. This clarifies the practical distinction between offline explanation calibration and online explanation generation.

Once the transition matrix is fitted, the runtime behavior of ETM is identical to that of the original transition-matrix explainer: one FM forward pass produces the deep feature vector and one matrix multiplication $b^* = Ta$ produces the explanation. In other words, the symmetry constraint changes training-time fitting cost but not deployment-time explanation latency. This distinction is important when ETM is compared with perturbation-heavy explainers such as LIME, SHAP variants, or robustness-oriented local surrogates, whose explanation cost scales with the number of samples or perturbations per query.

S11. Generator Verification and Lie Algebra Structure

Generator verification complements generator estimation. The goal is not to demand mathematically perfect Lie-algebra structure from noisy learned features, but to measure how far the empirical operators deviate from the expected structure and whether a simple projection step improves the downstream commutator penalty.

For a one-parameter Lie group such as $SO(2)$, there is only one generator, so Lie-algebra closure is vacuous. The more relevant check is whether the induced local action explains the observed finite differences well, which is exactly what η_A and η_B quantify. For multi-generator settings, the situation is richer: the generators should also respect the commutator relations of the chosen Lie algebra. In the controlled $SE(2)$ benchmark above, these relations are satisfied by construction. In real data settings, violation of those commutator relations can itself be used as a diagnostic that the selected group is only approximate.

S12. Diagnostic Tools for Symmetry Hypothesis Validation

Because ETM can over-regularize when the symmetry hypothesis is wrong, diagnostics are part of the method rather than an optional appendix. The quantities reported below provide concrete signals that a chosen group or MM may be misaligned with the learned representation.

A practical deployment question is how to decide whether a candidate symmetry group is appropriate before ETM is trusted. We recommend three checks. First, estimate the generator-fit residuals η_A and η_B; large values indicate that the chosen group does not linearize the data well in the selected FM/MM spaces. Second, inspect the discrete L-curve described in the main text; if the equivariance residual does not decrease smoothly as λ grows, the assumed symmetry may be inconsistent with the data. Third, compare qualitative trajectories of explanations under transformation. If the commutator norm decreases while the explanation becomes semantically implausible, the group action or the MM design should be reconsidered.

S13. Alternative Mental Model Spaces

The main paper uses pixels as the MM because that choice makes qualitative interpretation immediate. Section S13 broadens the discussion by considering concept-based MMs, which trade completeness for semantic compactness. The comparison is useful because it shows that ETM is not intrinsically tied to image reconstruction as the explanatory object.

The revised manuscript uses the pixel-space image as the MNIST MM because it yields a complete, directly visualizable explanation space. Nevertheless, ETM is not tied to pixel-space MMs. Concept vectors, clinical descriptors, or engineered domain attributes are equally possible, provided that the same symmetry can be meaningfully expressed in both FM and MM spaces. In fact, when a reviewer asks whether explanations remain interpretable to human users, the MM choice is often more decisive than the ETM solver itself. ETM improves structural consistency of whatever MM is chosen; it does not manufacture semantic meaningfulness if the MM features are poorly designed.

S14. Point-Cloud Scalability Note

The main manuscript briefly notes that ETM should extend naturally to 3D geometric domains when both the FM and MM support a coherent rigid-motion action. Point clouds are a representative example. Recent point-cloud architectures such as GPSFormer and DAF-Net combine global contextual reasoning with explicit local geometric fitting [1,2]. ETM could be layered on top of such backbones if the explanation target were defined as a set of interpretable geometric descriptors, canonicalized occupancy patterns, or reconstructed template-aligned attributes. The core bottleneck is not the ETM objective but the design of an MM whose response to SE(3)-type motions is both meaningful to users and estimable from data.

We intentionally do not report a full point-cloud experiment in this revision because doing so responsibly would require a nontrivial redesign of the MM, generator-estimation routine, and visualization protocol. Instead, the supplement positions point clouds as a concrete next empirical frontier: the mathematical machinery is compatible with multi-generator rigid-motion groups, but the explanatory semantics must be engineered with the same care as the predictive architecture.

S15. Code Implementation Details

The code fragments are included for transparency rather than as a substitute for the repository. They expose the exact operator-level logic used by the implementation and make explicit how the Kronecker identities are realized without materializing the full system matrix.

For transparency, the revised paper now ties each major methodological step to a concrete file in the released repository. Table S9 provides a compact map between manuscript components and the implementation files.

Table S9. Repository map for the released implementation.

File	Role in the workflow
`src/mnist/model.py`	frozen CNN backbone and penultimate feature representation
`src/mnist/generators.py`	central-difference estimation of J^A and J^B for the MNIST experiment
`src/mnist/transition.py`	baseline and ETM transition fitting, including the implicit LSQR operator
`src/synthetic/core.py`	synthetic MDS bridge, generator estimation, lambda sweep, and robustness analysis
`configs/mnist.yaml`, `configs/synthetic.yaml`	released hyperparameter settings

In addition, the synthetic matrices used in the controlled benchmark remain in Appendix A of the main manuscript, so the essential data required to reproduce the core mathematical example are available even without downloading the external repository.

References

1. Wang, C.; Wu, M.; Lam, S.K.; Ning, X.; Yu, S.; Wang, R.; Li, W.; Srikanthan, T. GPSFormer: A global perception and local structure fitting-based transformer for point cloud understanding. In Proceedings of the Computer Vision – ECCV 2024; Leonardis, A.; Ricci, E.; Roth, S.; Russakovsky, O.; Sattler, T.; Varol, G., Eds., Cham, 2025; Vol. 15066, *Lecture Notes in Computer Science*, pp. 75–92. https://doi.org/10.1007/978-3-031-73242-3_5.
2. Wang, C.; He, S.; Fang, X.; Han, J.; Liu, Z.; Ning, X.; Li, W.; Tiwari, P. Point clouds meets physics: Dynamic acoustic field fitting network for point cloud understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 22182–22192.