Supplementary Material: Proof provenance

for "Sequential Products on Euclidean Jordan Algebras: Classification in Rank at Least Three and the Complex Qubit"

These sections collect the provenance ledger and the machine-verification inventory supporting the main paper; they are not required for the paper proofs, which are self-contained in the main text.

S1 What Is Machine-Checked

The claims of the paper are established by the proofs in the main article. Two machine artifacts accompany them: a Lean 4 proof artifact and the SymPy script `verify_n2.py`. Neither is claimed to *fully formalize* the theorem; the proofs of record remain those in the main article.

The Lean skeleton. A Lean 4 development (Mathlib v4.28.0) is a *conditional dependency skeleton* of the proof of Theorem 1. From explicit, citable interface fields it machine-checks selected downstream algebraic implications: the typewise differential collapses, the parameter globalization, the rank-two algebraic core on concrete $M_2(\mathbb{C})^{\mathrm{sa}}$, and the central decomposition of a direct-sum product (Proposition 3.8). The skeleton is about 2,760 lines across the twelve `MasterTheorem/` modules, with zero live `sorry` by source inspection and the library target building clean. Its `master_chain` closure carries no custom axioms: `#print axioms` returns exactly Lean's three core axioms. Most cited inputs enter as explicit interface fields, and two cited external results are recorded as `axiom` declarations outside that closure (§S3).

The development *encodes* the seven effect-level clause signatures (S1)–(S7) in the `SequentialProduct` typeclass, with (S2) represented as continuity of $a \mapsto a \cdot b$ on effects in the carried norm, and freezes the two product-level conclusion shapes. But it is *not* a formalization of Theorem 1. It constructs no concrete full instance of (S1)–(S7), and it does not build the comparison character or its differential from Θ_a, state the master theorem as an equality of sequential products, or discharge (S2)-continuity for a concrete product. All of those remain with the paper proofs. In particular, the `#print axioms` closure reported here audits the Lean kernel dependencies of theorems *quantified over* these interface structures. It does not establish the interface fields themselves, nor certify that they faithfully instantiate the cited results; those correspondences are a matter of source and paper review (ledgered in §S3). The closure *does* guarantee that no step *between* the cited inputs introduces a hidden axiom or `sorry`.

The rank-two boundary is different in kind. On a fixed spectral frame Lean defines the operation $a \cdot b = F_a\, b\, F_a^{\dagger}$ concretely on $M_2(\mathbb{C})^{\mathrm{sa}}$ and proves, with no interface fields, a selected algebraic core. That core covers the block normal form with its scalar, rank-one, and zero-twist degenerations, positivity and effect preservation, and the reference and Hadamard dial values with their swap-invariance. It also covers the shared-frame cocycle, backward compatibility on commuting effects, and the generator-level rank-two necessity and exchange-covariance selector. This part is a machine-checked algebraic computation rather than a conditional skeleton. It does *not* formalize the complete (S1)–(S7) verification of the τ-family, the remaining cocycle and compatibility subcases, concrete (S2)-continuity, Lemmas 6.5 to 6.7, the general sufficiency result Proposition 6.3, or the assembled bijection Corollary 6.8. Those remain paper and SymPy arguments, itemized in §S3. The full interface-field inventory and itemized coverage report are in §S3 below and accompany the archived source.

The script `verify_n2.py`. The standalone `verify_n2.py` (SymPy [MSP$^+$17], exact symbolic arithmetic) corroborates the algebraic identities V1–V9 in the proofs of Proposition 6.3 and Theorem 6.4, together with one auxiliary constant V10 that is retained as a regression check but no longer used by the shortened Step 3. These checks corroborate the self-contained proofs of Section 6 rather than replace them, and are itemized in §S4 below. The interferometric detection formula of Remark 6.10 is an elementary 2×2 trace computation, checked by hand.

S2 The input ledger and nearest prior results

The input ledger for Theorem 1 is: van de Wetering's Propositions 3.9 (equivalently [vdW19, Prop. 2(3)]), 4.19/4.20/4.22/5.2/5.3/5.5/5.7 and the compatibility bridge [vdW18, vdW20]; the van Imhoff–Roelands order-isomorphism theorem [vIR20]; and the Spin(8) identification of [Yok09] for the exceptional type.

Hypotheses of the imported propositions. The hypotheses of each imported proposition are discharged at their point of use in Remark 3.11. The simplicity hypotheses among Propositions 4.19–4.20, 5.2, 5.5 and 5.7 are met because every effect of a finite-dimensional EJA is simple (Lemma 3.3(ii)). The order-preservation hypotheses of Propositions 4.22 and 5.3 (for 5.3, with respect to both products) are Lemma 3.3(iii), Proposition 5.7's reference-invariance is Lemma 3.3(iv), and the sole normality-type input is the scalar-homogeneity step of Lemma 3.1, where (S2) substitutes for σ-normality (Remark 3.5). Because none of these propositions requires the *ambient* algebra to be simple—only the effects involved—Proposition 3.8 applies the Proposition 5.2 transfer inside a *non-simple* direct sum with no extra hypothesis.

The nearest prior results are the following. Van de Wetering's Section 5 [vdW18] stops short of classifying the residual comparison automorphism. His invariance-based all-EJA uniqueness (his Theorem 5.11) forces Lüders on every type, but assumes norm-continuity in the first argument together with invariance of the *unknown* product; we do not assume the latter. Remark 3.12 of the main article discusses the exceptional case in that theorem's proof. There are also the purity/dagger selectors of [WWvdW19]. Finally, the reconstruction of Westerbaan and van de Wetering records an analogous assert-map uniqueness gap: their assert maps are unique only for simple invertible predicates (a norm-dense set), with full uniqueness recovered from diamond-positivity only once the predicate spaces are JBW-algebras [WvdW22, Remark 116, §5.3]. Van de Wetering's thesis characterizes the assert map on a JBW-algebra as the unique diamond-positive map f with $f(1) = a$ [vdW21, Thm. 4.6.17]; the Westerbaan–Westerbaan universal property [WW16] and the taxonomy of normal sequential effect algebras [WWvdW20] address broader axiomatization questions by adding categorical structure.

S3 The Lean skeleton: interface fields and coverage

Axioms and the classical-import surface. The `master_chain` skeleton declares no custom axioms; `#print axioms` on `master_chain` returns exactly Lean's three core axioms (`propext`, `Classical.choice`, `Quot.sound`). This is a syntactic axiom-closure figure, not a certificate of source fidelity: the classical inputs are carried as interface *fields*, which a reviewer audits by reading the structure definitions rather than as global axioms. For the same reason the development constructs no witness of any of those interface structures, so a green build does not certify that they are inhabited. The audit freezes their constructors against field drift. But a constructor type names its helper predicates opaquely, so a body-level edit to such a predicate could in principle render the skeleton vacuous while leaving every frozen type and axiom closure unchanged. The archive's audit file records this limit at that point. Nothing in the paper's proofs depends on the skeleton.

There are three classical imports. The first is van Imhoff–Roelands, Corollary 2.5 [vIR20], supporting the Proposition 3.10 upgrade; it enters as the field `Θ_jordan`, a hypothesis scoped to the instances a caller constructs, since the interface does not itself encode the JB-algebra premises. The second is the Spin(8)-triality faithfulness recorded in [Yok09] (the field `IsAlbertModel.block_injective`). The third is the standard smoothness of a continuous Lie-group homomorphism, whose additive-to-linear upgrade is *proved* (Mathlib's `AddMonoidHom.toRealLinearMap`), with only the continuity surviving as a field (`dχAdd_cont`). Uniqueness of continuous characters of $\mathbb{R}$ is *proved* (`real_character_unique`), not assumed. The van de Wetering comparison propositions (4.20, 5.2, 5.3, 5.5, 5.7) and his EJA-appendix compatibility propositions are likewise interface fields. The differential face of the setup consumed by the diagonal-character constructor comprises, in full, the block-action maps ρ_{ij} (the paper-intended isotropy representations) and their quadratic skewness (ρ_{skew}), the additive differential `dχAdd` together with its continuity `dχAdd_cont`, and `coalescence_diff`. Lean carries all five as fields: it neither constructs `dχAdd` from the proved comparison cocycle nor equates it with a derivative of Θ. The comparison-to-differential transition is therefore supplied, not machine-checked.

What the skeleton machine-checks. From these fields the development checks:

- the hyperplane factorization;
- the coupling identity as a conclusion of the constructor, not a hypothesis;
- three conditional typewise $T = 0$ results (real, rank-independent; quaternionic, two-slot with $Z(\mathbb{H}) \cap \operatorname{Im} \mathbb{H} = \{0\}$ computed; exceptional, from the cited block injectivity), together with the complex per-frame scalar collapse;
- character equality on an interval implying exact parameter equality;
- the frame-connectivity induction, with the geometric two-plane rotation move a located hypothesis;
- a generic dense-agreement lemma stated standalone, not invoked by the capstone;
- the globalization family anchored to the produced coupling by an explicit hypothesis.

Separately, on concrete $M_2(\mathbb{C})^{\mathrm{sa}}$, it checks the rank-two algebraic core: the block form, the Lüders value at $t = 0$ and the degenerations, the τ frame values, the shared-frame cocycle, and backward compatibility. It also checks, at generator level, the exchange-covariance selector of Remark 6.2, though not the full rank-two boundary theorem. This rank-two core is verified for an *arbitrary* real twist parameter, exactly the frame-independent algebraic content invoked by Proposition 6.3. The central decomposition (Proposition 3.8) is machine-checked as the componentwise identity $a \cdot b = \sum_\alpha a_\alpha \cdot b_\alpha$, the proof consuming only (S1) and (S5) with the bridge and Proposition 5.2 facts carried as hypotheses; its summand-inheritance clause (that each restriction is itself a sequential product) remains a paper proof.

What the skeleton does not check. The development does not check:

- the construction or verification of a concrete EJA operation as an instance of the (S1)–(S7) interface, in particular discharging (S2)-continuity for a concrete product;
- the construction of the comparison character and its differential from Θ_a;
- the operator-to-character step of the cross-coherence overlap;
- the geometric connectivity move;
- (S2)-continuity and invertible-density, the hypotheses of the standalone dense-agreement lemma;
- at rank two, the unimodular subcases of the Step-6 cocycle and the bundled seven-axioms verification of the τ family;
- the continuity, boundedness, and descent of the rank-two frame parameter (Lemmas 6.5 to 6.7), the assembled rank-two classification (Corollary 6.8), and the frame-function packaging and (S2)-continuity of Proposition 6.3, all resting on $\mathbb{RP}^2$ topology and matrix-logarithm analysis outside the development;
- the finite-dimensional omnibus classification (Theorem 2), the pseudo-inverse transfer (Proposition 3.9 of the main article), and the sufficiency lemma for the constant-twist family (Lemma 5.10), none of which has any counterpart in the development;
- the elementary order-unit and simple-effect lemmas (Lemmas 2.3, 3.1 to 3.4 and 3.7), Lemma 5.4, and Corollary 7.1;
- the cited van de Wetering propositions' own contents.

Theorem 1 in print rests on the paper proofs; the Lean artifact checks the downstream skeleton from cited fields. A separate earlier Lean development (`SequentialProductsArtifact/Selection/`) machine-checks the algebraic block core of the pair-local ansatz route of Section 8 (the block normal form, the equivalence $T = 0 \iff$ Lüders $\iff$ exchange covariance, and the rank-two isotropy calculation) and is retained unchanged. Its existence legs rest on the operator form of the classical multiplicative-continuity theorem (the scalar case is Aczél [Acz66]; the operator upgrade is standard one-parameter-group theory, e.g. [EN00, Thm. I.2.9]), declared as an axiom in `Selection/NormalFormExistence.lean`. The standalone tree separately declares the Barnum–Graydon–Wilce composite table [BGW20] (in `TwistNormalForm.lean`, supporting its retained self-composition material). Both axioms lie outside the `master_chain` import tree and do not enter its axiom closure.

S4 The SymPy identities `verify_n2.py` (V1–V10)

The identities labelled V1–V9 in the proofs of Proposition 6.3 and Theorem 6.4, together with one auxiliary constant V10, are corroborated by the standalone `verify_n2.py` (SymPy [MSP+17], exact symbolic arithmetic, no floating point). The identities are the block form of the frame-dependent family (9) and its degenerations (V1–V3), the trace identities behind the compatibility criterion (V4), the phase cocycle (V5). The rest are (S5) across the compatible cases (V6), the backward compatibility direction (V7), (S4) instances (V8), and the frame values of τ (V9). The auxiliary constant $\sup_{z\in(0,1]} \sqrt{z}|\log z| = 2/e$ (V10) is retained as a regression check but is no longer used by the shortened Step 3, which bounds the $\lambda_0 = 0$ limit directly by $0 \le a \cdot b \le a$.

The script itself prints its groups as `1.`–`10c.` without the `V` prefix: label Vk above is script group k, and the lettered sub-labels (`4a`, `5b`, …) are the individual checks within a group. A green run emits 33 `PASS` lines and exits 0.

References

[Acz66] János Aczél. *Lectures on Functional Equations and Their Applications*, volume 19 of *Mathematics in Science and Engineering*. Academic Press, 1966.

[BGW20] Howard Barnum, Matthew A. Graydon, and Alexander Wilce. Composites and categories of Euclidean Jordan algebras. *Quantum*, 4:359, 2020. arXiv:1606.09331.

[EN00] Klaus-Jochen Engel and Rainer Nagel. *One-Parameter Semigroups for Linear Evolution Equations*, volume 194 of *Graduate Texts in Mathematics*. Springer, 2000.

[MSP+17] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, et al. SymPy: symbolic computing in Python. *PeerJ Computer Science*, 3:e103, 2017.

[vdW18] John van de Wetering. Three characterisations of the sequential product. *Journal of Mathematical Physics*, 59(8):082202, 2018. arXiv:1803.08453.

[vdW19] John van de Wetering. Sequential product spaces are Jordan algebras. *Journal of Mathematical Physics*, 60(6):062201, 2019. arXiv:1803.11139.

[vdW20] John van de Wetering. Commutativity in Jordan operator algebras. *Journal of Pure and Applied Algebra*, 224(11):106407, 2020. arXiv:1912.01903.

[vdW21] John van de Wetering. *Quantum Theory from Principles, Quantum Software from Diagrams*. PhD thesis, Radboud University Nijmegen, 2021. arXiv:2101.03608.

[vIR20] Hendrik van Imhoff and Mark Roelands. Order isomorphisms between cones of JB-algebras. *Studia Mathematica*, 254:179–198, 2020. arXiv:1904.09278.

[WvdW22] Bas Westerbaan and John van de Wetering. A computer scientist's reconstruction of quantum theory. *Journal of Physics A: Mathematical and Theoretical*, 55(38):384002, 2022. arXiv:2109.10707.

[WW16] Abraham Westerbaan and Bas Westerbaan. A universal property for sequential measurement. *Journal of Mathematical Physics*, 57(9):092203, 2016. arXiv:1603.00410.

[WWvdW19] Abraham Westerbaan, Bas Westerbaan, and John van de Wetering. Pure maps between Euclidean Jordan algebras. *Electronic Proceedings in Theoretical Computer Science*, 287:345–364, 2019. arXiv:1805.11496.

[WWvdW20] Abraham Westerbaan, Bas Westerbaan, and John van de Wetering. The three types of normal sequential effect algebras. *Quantum*, 4:378, 2020. arXiv:2004.12749.

[Yok09] Ichiro Yokota. Exceptional Lie groups, 2009. arXiv:0902.0431, 204 pp. monograph.