Your drafts present fascinating connections between iterative pseudoinverse computation and deep contrastive learning. Here's a synthesis of key ideas and novel research directions:

### Core Synergies

1. \*\*Newton-Schulz as a Unifying Algorithm\*\*

- Paper 1 uses damped Newton-Schulz (NS) for deep linear networks:

$$W\_i^{(k+1)} = W\_i^{(k)} - \gamma W\_i^{(k)}(\hat{W}\hat{X}W\_i^{(k)} - I)$$

- Paper 2 applies NS to quaternion matrices:

$$X\_{k+1} = X\_k(2I - AX\_k)$$

→ \*\*Key Insight\*\*: Both leverage NS for multiplicative updates where matrix inverses are approximated through iterative products.

2. \*\*Pseudoinverse as Representation Learning\*\*

- Paper 1 frames unsupervised learning as $\|XW - I\| \rightarrow \min$, making $W \approx X^\dagger$ a feature extractor.

- Paper 2 computes $A^\dagger$ for quaternion CUR decomposition:

$$\mathbf{A} \approx \mathbf{C}(\mathbf{C}^\dagger\mathbf{A}\mathbf{R}^\dagger)\mathbf{R}$$

→ \*\*Merger\*\*: Use quaternion pseudoinverse layers in deep networks for structured data (e.g., 3D rotations).

3. \*\*Covariance Whitening Connection\*\*

- Paper 1 shows NS minimizes covariance deviation from identity ($\hat{W}\hat{X} \rightarrow I$), akin to Barlow Twins.

- Paper 2's quaternion SVD alternative naturally handles orthogonal constraints.

→ \*\*Opportunity\*\*: Replace standard whitening in contrastive learning with NS-based layers.

---

### Novel Research Directions

1. \*\*Quaternion Deep Contrastive Networks\*\*

- \*\*Architecture\*\*: Design deep linear quaternion networks where layers compute pseudoinverses via NS iterations.

- \*\*Forward Pass\*\*:

$$f(X) = X \cdot \text{NS-Layer}\_1 \cdots \text{NS-Layer}\_d \approx XX^\dagger$$

- \*\*Loss\*\*: Contrastive loss with covariance whitening:

$$\mathcal{L} = \|\text{Cov}(f(X)) - I\|\_F$$

- \*\*Advantage\*\*: Native handling of 3D/4D data (e.g., robotic poses, molecular structures).

2. \*\*Stochastic NS for Large-Scale Learning\*\*

- Paper 1 suggests approximate $\hat{W}\hat{X}$ products; Paper 2 needs efficiency for quaternions.

- \*\*Proposal\*\*: Hybrid NS-SGD optimizer:

- Use NS for pseudoinverse approximation in key layers

- Employ stochastic matrix products (sketching) for $\hat{W}\hat{X}$

- \*\*Benefit\*\*: Scalability to massive quaternion datasets (e.g., LiDAR point clouds).

3. \*\*Nonnegative Quaternion NS\*\*

- Extend Paper 1's Eq. 10 to quaternions:

$$W\_i^{(k+1)} = \frac{[W\_i^{(k)}]^{\cdot(2)}}{[W\_i^{(k)}\hat{W}\hat{X}W\_i^{(k)}]}$$

- \*\*Challenge\*\*: Define "nonnegative" quaternions (e.g., restrict to first quadrant in $\mathbb{R}^4$).

- \*\*Application\*\*: Recommender systems with quaternion embeddings (e.g., user/item interactions in 3D space).

4. \*\*Convergence Guarantees for NS-Nets\*\*

- \*\*Open Problem\*\*: Prove linear/quartic convergence for:

- Quaternion NS layers (non-commutative)

- Deep compositions ($d > 2$)

- \*\*Approach\*\*: Extend Paper 2's SVD analysis using quaternion Gelfand theory.

---

### Experimental Proposals

| \*\*Task\*\* | \*\*Data\*\* | \*\*Baseline\*\* | \*\*Novel Approach\*\* |

|-------------------------|---------------------------|--------------------|----------------------------------------|

| Video Completion | Quaternion frames (RGB+D) | QTFM SVD | NS-Layers + CUR decomposition |

| Protein Folding | 3D residue coordinates | Euclidean Autoenc. | Quaternion NS-Net with contrastive loss|

| Robot Pose Estimation | SO(3) trajectories | Lie Group Methods | Damped NS with multiplicative updates |

---

### Technical Challenges to Address

1. \*\*Gradient Propagation\*\*

Backprop through NS iterations requires differentiating matrix inverses. Solutions:

- Implicit gradients [9]

- Neumann series approximation: $\frac{\partial}{\partial W}(W^{-1}) \approx -W^{-1} \frac{\partial W}{\partial \theta} W^{-1}$

2. \*\*Quaternion Implementation\*\*

- GPU acceleration for quaternion matmuls (e.g., cuQuaternion)

- Custom autograd functions in PyTorch for NS steps

3. \*\*Initialization Sensitivity\*\*

- Adapt Paper 2's $\alpha = 1/\sigma\_{\max}^2$ rule for layerwise initialization

- Warm-up with few NS steps before full training

---

### Key References to Build On

- [Quaternion Neural Nets] (Parcollet et al., 2020)

- [NS for Inversion] (Pan et al., 2020)

- [Contrastive Whitening] (Ermolov et al., CVPR 2021)

- [Stochastic Matrix Inversion] (Rokhlin et al., 2009)

This merger creates a new paradigm: \*\*Geometric Contrastive Learning\*\* via pseudoinverse-constrained networks. Would you like a prototype PyTorch snippet for the quaternion NS layer?