

# NetEV Lookup Table Gap Fix - Step-by-Step Action Plan

## Current Situation

- **665 existing base rows** before extrapolation
- **Major gaps** in essential hand types (4K, pairs, flushes)
- **Extrapolation creating invalid tuples** and hierarchy violations
- **NetEV losing competitive precision** due to missing kicker distinctions

## Goal

- **Complete, valid lookup table** with no gaps
  - **Proper hierarchy preservation** across all hands
  - **Multi-position storage** for efficiency
  - **Strategic precision** for competitive distinctions (8-point deltas)
- 

## PHASE 1: ANALYSIS & PLANNING

### Step 1: Categorize Current 665 Rows

**Action:** Analyze what we actually have vs what we need

- ☐ Group current rows by hand type and element count
- ☐ Identify which hand types use 2-element vs 3-element
- ☐ Document current coverage gaps by hand type
- ☐ Create "Current State" summary document

**Deliverable:** Complete inventory of existing coverage

### Step 2: Define Target Schema

**Action:** Establish the final lookup table structure

- ☐ Confirm 2-element hand types: 5K, 6K, 7K, 8K, Front Trips
- ☐ Confirm 3-element hand types: 4K, Full House, Pairs, Straights, SFs, Flushes
- ☐ Define valid tuple patterns for each hand type
- ☐ Document multi-position storage format

**Deliverable:** Target schema specification document

### Step 3: Calculate Target Row Counts

**Action:** Determine exactly how many rows we need

- ☐ Calculate valid combinations per hand type (with constraints)
- ☐ Account for "kickers cannot improve hand type" rule
- ☐ Sum total target rows needed
- ☐ Identify priority hand types (biggest gaps/most strategic value)

**Deliverable:** Target row count by hand type

---

## PHASE 2: GAP IDENTIFICATION

### Step 4: Systematic Gap Analysis

**Action:** Find exactly what's missing

- ☐ For each hand type, list all valid tuples that should exist
- ☐ Compare against current 665 rows
- ☐ Create "missing tuple" lists by hand type
- ☐ Prioritize gaps by strategic importance

**Deliverable:** Comprehensive gap list with priorities

### Step 5: Validate Gap Lists

**Action:** Ensure gap lists contain only valid poker hands

- ☐ Apply poker constraints (straights, valid kickers, etc.)
- ☐ Remove impossible combinations
- ☐ Verify hierarchy relationships
- ☐ Double-check against hand type rules

**Deliverable:** Validated missing tuple lists

---

## PHASE 3: TARGETED GAP FILLING

### Step 6: Expand High-Priority Hand Types

**Action:** Start with most strategically important gaps **Priority Order:**

1. **Four of a Kind** (expand from 13 → 156 tuples)
2. **Pairs** (expand from 13 → 156 tuples)
3. **Full House** (expand from 13 → 156 tuples)

4. **Flushes** (expand from 11 → 156 tuples)

**For each hand type:**

- ☐ Generate all valid 3-element tuples
- ☐ Use bounded extrapolation (hierarchy neighbors)
- ☐ Validate hierarchy preservation
- ☐ Test with sample NetEV queries

**Deliverable:** Expanded lookup tables for priority hand types

## Step 7: Complete Remaining Hand Types

**Action:** Fill remaining gaps systematically

- ☐ Straights and Straight Flushes (validate sequences)
- ☐ Two Pair (rationalize current 404 → target ~78)
- ☐ Same rank hands (7K, 8K completion)
- ☐ Any remaining minor gaps

**Deliverable:** Complete lookup table draft

---

## PHASE 4: VALIDATION & TESTING

### Step 8: Hierarchy Validation

**Action:** Ensure no violations across entire table

- ☐ Sort all hands by strength within positions
- ☐ Verify EV values maintain proper order
- ☐ Fix any hierarchy violations
- ☐ Document validation results

**Deliverable:** Hierarchy-validated lookup table

### Step 9: NetEV Integration Testing

**Action:** Test with real NetEV queries

- ☐ Run problematic test cases (1001, 1002, 1003)
- ☐ Verify strategic decisions improve
- ☐ Check for missing lookups during execution
- ☐ Validate competitive distinctions work

**Deliverable:** Tested, working NetEV system

## Step 10: Performance Validation

**Action:** Re-run full test suite

- ☐ Execute all 30 test cases with new lookup table
- ☐ Compare NetEV performance vs other methods
- ☐ Document strategic improvements
- ☐ Verify bug fixes resolved

**Deliverable:** Complete performance analysis

---

## SUCCESS CRITERIA

### Technical Success

- ☐ Zero invalid poker hand tuples
- ☐ Perfect hierarchy preservation
- ☐ Complete coverage for algorithm needs
- ☐ No extrapolation errors

### Strategic Success

- ☐ NetEV makes superior strategic choices
  - ☐ Competitive distinctions captured (8-point deltas)
  - ☐ Test performance improvements documented
  - ☐ Strategic reasoning validates design
- 

## RISK MITIGATION

### Backup Current State

- ☐ Save complete current lookup table
- ☐ Document current NetEV test results
- ☐ Create rollback procedures

### Incremental Validation

- ☐ Test each hand type expansion separately
- ☐ Validate hierarchy after each addition
- ☐ Check NetEV behavior after each phase

## Quality Gates

- ☐ No invalid tuples pass validation
  - ☐ Hierarchy checked at every step
  - ☐ NetEV performance monitored throughout
- 

## TIMELINE ESTIMATE

- **Phase 1:** 2-3 sessions (analysis & planning)
- **Phase 2:** 1-2 sessions (gap identification)
- **Phase 3:** 3-4 sessions (gap filling)
- **Phase 4:** 2-3 sessions (validation & testing)

**Total:** 8-12 focused work sessions

---

## NEXT SESSION FOCUS

**Immediate Action:** Start with Step 1 - Categorize Current 665 Rows

- Analyze hand type distribution in real data
- Identify 2-element vs 3-element patterns
- Document biggest coverage gaps
- Set foundation for systematic gap filling