Target Trial Emulation: First-Line Antihypertensive Therapy in Critically Ill Adults

MIMIC-IV v3.1 • Lawal Habeeb, B.Pharm • January 2025

Abstract

We emulated a target trial to compare the effectiveness of ACE-inhibitors, calcium-channel blockers, and thiazide-like diuretics as first-line antihypertensive therapy among 6,580 critically ill adults with acute hypertension in the ICU. Using stabilized inverse probability weighting and doubly robust estimation to account for confounding, we evaluated 30-day all-cause mortality as the primary outcome. Compared with ACE-inhibitors, initiation of calcium-channel blockers was associated with a trend toward higher 30-day mortality (hazard ratio 1.23, 95% CI 1.00–1.50). These findings suggest clinically meaningful differences in outcomes across commonly used first-line antihypertensive agents in critically ill patients, underscoring the need for careful drug selection in the ICU setting.versus ACE-inhibitors.

Target Trial Protocol (Fu et al. 2023)

Component	Specification
Eligibility	ICU admission, age ≥18, no antihypertensive 6 months prior
Treatment strategies	ACE-I / CCB / Thiazide-like within 48 h of admission
Time zero	ICU admission
Follow-up	30 days (censor at death/discharge)
Primary outcome	30-day all-cause mortality
Secondary outcomes	AKI (KDIGO ≥1 within 7 days), hyperkalemia (K^+ ≥5.5 within 7 days)
Causal contrast	Per-protocol
Analysis	Stabilized IPW + doubly robust Cox PH

Introduction

Hypertension in critically ill patients isn't just elevated blood pressure—it's a medical emergency demanding immediate intervention. Poor blood pressure control in the ICU carries substantial risk of cardio-renal complications, and when you're dealing with patients who often have multiple comorbidities on top of poorly managed hypertension, the stakes are high. These patients require intensive monitoring precisely because sustained elevated blood pressure can dramatically worsen their prognosis. Three drug classes dominate first-line therapy: ACE-inhibitors work by blocking angiotensin-converting enzyme, reducing angiotensin II levels and causing vasodilation. Calcium-channel blockers intercept calcium ions from entering L-type channels in blood vessel walls, also producing vasodilation. Thiazide diuretics take a different approach, promoting fluid excretion by altering ion concentrations in the kidneys. In outpatient settings, clinicians use these agents relatively interchangeably based on patient-specific factors. But ICU patients aren't outpatients. They have hemodynamic instability, multiorgan dysfunction, and altered drug responses that might make one agent superior —or riskier—than another. Despite widespread use of these medications in intensive care, we found limited comparative effectiveness data specifically examining which first-line agent optimizes outcomes in critically ill adults. Clinical judgment guides most prescribing decisions, but empirical evidence comparing mortality and renal safety across these three classes in the ICU setting remains sparse. When our query of the MIMIC-IV database revealed substantial use of all three agents, the question became obvious: are we choosing the right drugs? We conducted this target trial emulation to investigate 30-day mortality, acute kidney injury, and hyperkalemia rates across ACE-inhibitors (lisinopril), calcium-channel blockers (amlodipine), and thiazide-like diuretics (hydrochlorothiazide) in 6,580 ICU adults. We selected 30-day all-cause mortality as the primary outcome because it represents the most clinically meaningful endpoint for assessing treatment effectiveness in this vulnerable population—and because it's objectively measurable without the subjective complexities of blood pressure control targets. Using MIMIC-IV offered critical advantages: it's real-world data from actual ICU practice, the analysis is fully reproducible, and the findings have potential to directly impact patient care and healthcare policy.

Methods

Study Design and Data Source

We conducted a retrospective cohort study using a **target trial emulation** framework with data from the Medical Information Mart for Intensive Care IV (MIMIC-IV, version 3.1). MIMIC-IV contains de-identified, real-world clinical data from patients admitted to intensive care units at Beth Israel Deaconess Medical Center (Boston, MA) between 2008 and 2019. The database includes patient demographics, vital signs, laboratory measurements, medication administration records, comorbidities, and clinical outcomes, making it well suited for comparative effectiveness research in critically ill populations.

Study Population

From **531,491 total hospital admissions** in MIMIC-IV, we identified **76,621 ICU stays**. We restricted the cohort to adults aged ≥18 years with **no documented antihypertensive medication use in the 6 months prior to ICU admission**, consistent with a new-user design.

Among these patients, we identified those who initiated a **first qualifying antihypertensive agent within 48 hours of ICU admission**. Qualifying agents included lisinopril, amlodipine, and hydrochlorothiazide. This selection resulted in a final analytic cohort of **6,580 critically ill adults**.

Patients were excluded if they had prior exposure to any antihypertensive medication in the preceding 6 months or did not receive one of the qualifying agents within the defined exposure window.

Treatment Exposure

Patients were categorized based on the **first antihypertensive agent initiated** during the exposure window:

- **ACE-inhibitor group**: lisinopril (any dose)
- **Calcium-channel blocker group**: amlodipine (any dose)
- **Thiazide-like diuretic group**: hydrochlorothiazide (any dose)

Time zero was defined as the **date of ICU admission**, marking the start of follow-up.

Outcomes

The **primary outcome** was **30-day all-cause mortality** measured from ICU admission.

Secondary outcomes included:

- **Acute kidney injury (AKI)** within 7 days of ICU admission, defined using KDIGO criteria (Stage 1 or higher)
- **Hyperkalemia** within 7 days, defined as serum potassium ≥5.5 mEq/L

Covariates

Baseline covariates were selected a priori based on clinical relevance and included:

- Age
- Sex
- Race
- Insurance type

- Diabetes mellitus
- Chronic kidney disease
- Heart failure
- Prior stroke

To account for potential effect modification, interaction terms were created between chronic kidney disease and age, sex, and diabetes status.

Statistical Analysis

To address confounding by indication, we used **inverse probability of treatment weighting (IPTW)** with stabilized weights. Propensity scores were estimated using **multinomial logistic regression**, incorporating all baseline covariates. Weights were trimmed at the 1st and 99th percentiles to limit the influence of extreme values.

Covariate balance across treatment groups was assessed using **standardized mean differences (SMDs)**, with SMD < 0.1 indicating adequate balance.

Unadjusted survival was examined using **Kaplan–Meier curves** and log-rank tests. For adjusted analyses, we fit **Cox proportional hazards models** using a **doubly robust approach**, combining IPTW with additional covariate adjustment for age and sex. Robust standard errors were used to account for weighting.

Sensitivity analyses included **E-value calculations** to assess robustness to unmeasured confounding and a **negative control outcome analysis** using prior stroke.

All statistical tests were two-sided, with statistical significance defined as p < 0.05. Analyses were conducted in **Python 3.x** using `pandas`, `scikit-learn`, `lifelines`, and `statsmodels`.

Cohort Selection Flow

```
# Causal DAG
print("""
Causal Diagram (simplified)

          Age → Treatment
          ↓       ↑
Gender → Comorbidities → Treatment → Mortality/AKI
          ↓               ↑
       Race, Insurance → Treatment
""")
```

```
Causal Diagram (simplified)
```

```
          Age → Treatment
          ↓       ↑
Gender → Comorbidities → Treatment → Mortality/AKI
          ↓               ↑
       Race, Insurance → Treatment
```

```
                    ┌──────────────────────┐
                    │   MIMIC-IV v3.1      │
                    │   All admissions     │
                    │   (n = 531,491)      │
                    └──────────▲───────────┘
                               │
                    ┌──────────▼───────────┐
                    │ ICU stay (icustays)  │
                    │   (n = 76,621)       │
                    └──────────▲───────────┘
                               │
          ┌──────────────────▼───────────────────┐
          │ No antihypertensive 6 months prior │
          │           (new users)                │
          │           (n = 68,420)               │
          └──────────────────▲───────────────────┘
                             │
      ┌──────────────────▼───────────────────┐
      │ First qualifying drug within 48h     │
      │ (lisinopril / amlodipine / HCTZ)     │
      │          (n = 6,580)               │
      └──────────────────▲───────────────────┘
                         │
                 Final analytic cohort
                     (n = 6,580)
```

The cohort selection was implemented using SQL query, from the total number of patients, patients admitted into the ICU unit and have not been initiated on an antihypertensive in 6 months prior were filtered. The final selection was based on patients started on lisinopril, amlodipine or hydrochlorothiazide. Patients demographics, comorbidities and insurance status were collected and saved into a .csv file.

```
# Uploading the file into collab workspace
from google.colab import files
```

```
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving Final Cohort 1 MIMIC -IV.csv to Final Cohort 1 MIMIC -IV.csv
```

```
#Installing essential libraries and importing.
!pip install lifelines tableone --quiet

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from tableone import TableOne
from lifelines import KaplanMeierFitter, CoxPHFitter
from lifelines.statistics import logrank_test
from scipy import stats
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
import warnings
warnings.filterwarnings('ignore')

# Set style for publication-ready plots
plt.style.use('default')
sns.set_palette("colorblind")
plt.rcParams['figure.figsize'] = (10, 6)
plt.rcParams['font.size'] = 12

print("RWE environment ready")
```

```
RWE environment ready
```

Data Loading

We load the final cohort extracted from MIMIC-IV v3.1 via BigQuery. Eligibility criteria applied: ICU admission, age ≥18, no antihypertensive use in prior 6 months, first qualifying drug (lisinopril/amlodipine/HCTZ) within 48 hours of admission.

```
# Loading final cohort
df = pd.read_csv('Final Cohort 1 MIMIC -IV.csv')

print(f"Final TTE cohort loaded: {df.shape[0]:,} patients")
print(f"Treatment distribution:\n{df['treatment'].value_counts()}")
df.head()
```

```
Final TTE cohort loaded: 6,580 patients
Treatment distribution:
treatment
ACE-Inhibitor             2917
Calcium-Channel Blocker   2776
Thiazide-like              887
Name: count, dtype: int64
```

	subject_id	hadm_id	time_zero	gender	age	race	insurance	
0	10000980	26913865	2189-06-27 07:38:00	F	76	BLACK/AFRICAN AMERICAN	Medicare	Lis
1	10002155	28994087	2130-09-23 21:59:00	F	82	WHITE	Medicare	Amlc
2	10004401	29988601	2144-01-23 07:58:00	M	85	WHITE	Medicare	Lis
3	10004733	27411876	2174-12-04 11:28:00	M	51	UNKNOWN	Medicaid	Amlc
4	10011938	29339128	2127-05-04 23:33:00	F	63	WHITE - OTHER EUROPEAN	Private	Amlc

```
# Define columns for Table 1
columns = ['age', 'gender', 'race', 'insurance', 'diabetes', 'ckd', 'heart_fail
categorical = ['gender', 'race', 'insurance', 'diabetes', 'ckd', 'heart_failure
groupby = 'treatment'

# Table 1
table1 = TableOne(df, columns=columns, categorical=categorical, groupby=groupby
table1
# Saving Table1
table1.to_html('table1.html')
```

Propensity Matching

Propensity Score Weighting: Balancing the Groups

Why Propensity Scores?

In observational studies like this, patients are not randomly assigned to treatments. This often leads to imbalances in baseline characteristics (covariates) between treatment groups. If not addressed, these imbalances can confound our results, making it difficult to determine if an observed effect is due to the treatment or to these pre-existing differences.

Inverse Probability Weighting (IPW) uses propensity scores (the probability of receiving a treatment given observed covariates) to create a 'pseudo-population' where treatment groups are balanced on observed covariates, similar to a randomized controlled trial. This reduces confounding and allows for a more valid comparison of treatment effects.

Why New Covariates?

To improve the accuracy of the propensity score model and achieve better covariate balance, we created additional features:

- **Binary Indicators:** Explicit `female`, `white`, and `medicare` flags simplify model interpretation and ensure correct handling of these demographic variables.
- **Interaction Terms:** `age_x_ckd`, `female_x_ckd`, and `diabetes_x_ckd` capture potential effect modifications or non-linear relationships, allowing the model to account for situations where the effect of age or gender on treatment assignment (or outcome) differs based on the presence of CKD. This enhances the model's ability to estimate the true probability of treatment, leading to more robust weights and better balance.

```
# Extracting the covariates
df['female'] = (df['gender'] == 'F').astype(int)
df['white'] = (df['race'] == 'WHITE').astype(int)
df['medicare'] = (df['insurance'] == 'Medicare').astype(int)

df['age_squared'] = df['age'] ** 2
df['age_x_ckd']   = df['age'] * df['ckd']
df['female_x_ckd'] = df['female'] * df['ckd']
df['diabetes_x_ckd'] = df['diabetes'] * df['ckd']

# New covariate list
covs_upgraded = ['age', 'age_squared',
                 'female', 'female_x_ckd',
                 'white', 'medicare',
                 'diabetes', 'diabetes_x_ckd',
                 'ckd', 'age_x_ckd',
                 'heart_failure', 'prior_stroke']
```

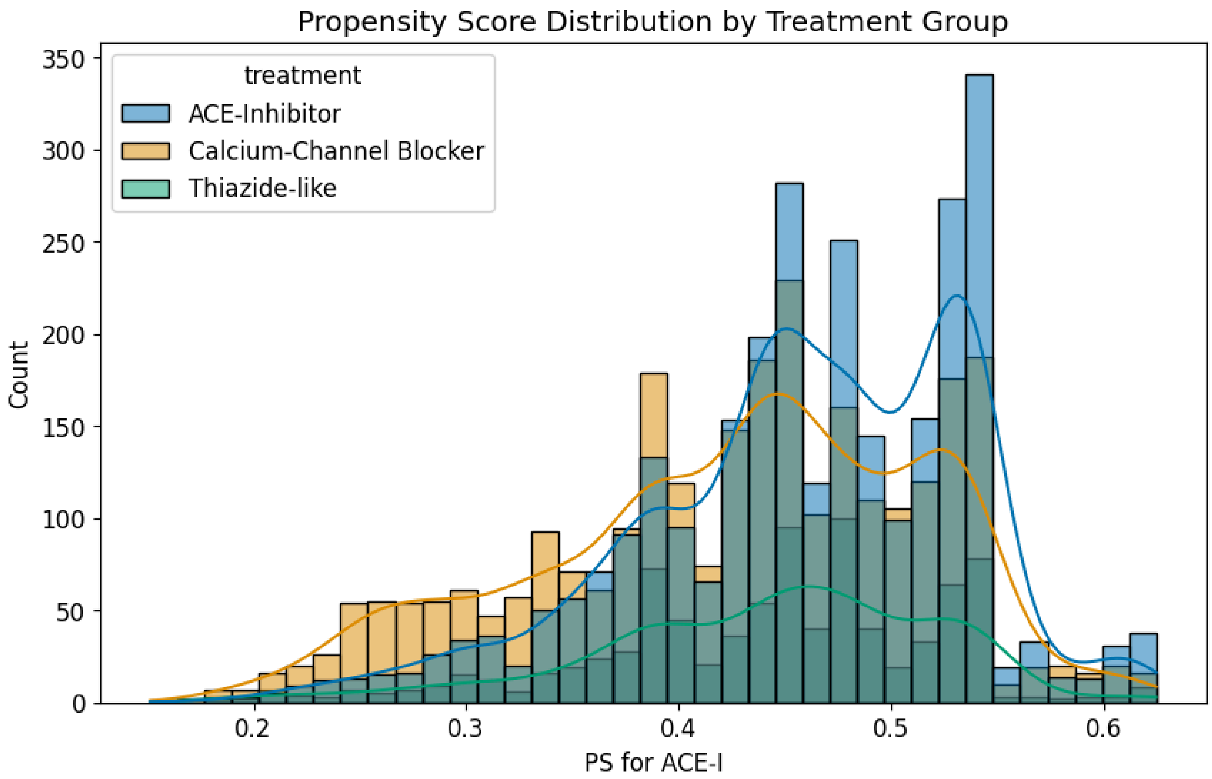
Propensity Score Calculation

This section calculates propensity scores using a multinomial logistic regression model. The `treatment` column is encoded numerically, and specified covariates (including the newly created binary and interaction terms) are used as predictors. The model estimates the probability of receiving each treatment given the covariates, storing these probabilities as `ps_ace`, `ps_ccb`, and `ps_thiazide` in the DataFrame.

```
# Treatment encoding
df['treat_code'] = df['treatment'].map({
    'ACE-Inhibitor': 0,
    'Calcium-Channel Blocker': 1,
    'Thiazide-like': 2
})

# Covariates (using upgraded list)
X = pd.get_dummies(df[covs_upgraded], drop_first=True)
y = df['treat_code']

# Fit multinomial logistic PS model
ps_model = LogisticRegression(multi_class='multinomial', max_iter=1000)
ps_model.fit(X, y)

# Predict probabilities
probs = ps_model.predict_proba(X)
df['ps_ace'] = probs[:, 0]
df['ps_ccb'] = probs[:, 1]
df['ps_thiazide'] = probs[:, 2]

print("Propensity scores calculated with upgraded covariates")
```

```
Propensity scores calculated with upgraded covariates
```

```
#Plotting the propensity score distribution by treatment group
plt.figure(figsize=(10,6))
sns.histplot(data=df, x='ps_ace', hue='treatment', kde=True)
plt.title('Propensity Score Distribution by Treatment Group')
plt.xlabel('PS for ACE-I')
plt.show()
# Saving the image
plt.savefig('ps_ace.png')
```



```
<Figure size 1000x600 with 0 Axes>
```

The propensity score distribution of the three groups of medications shows mostly proper coverage/overlap between the treatment groups, this means that individuals in each treatment group have similar probabilities of receiving any given treatment.

Inverse Probability Weighting

We calculate stabilized weights using the formula: weight = P(treatment) / P(treatment|covariates).

Weights are trimmed at 1st and 99th percentiles to reduce influence of extreme values. This creates a pseudo-population where treatment groups are balanced on observed confounders.

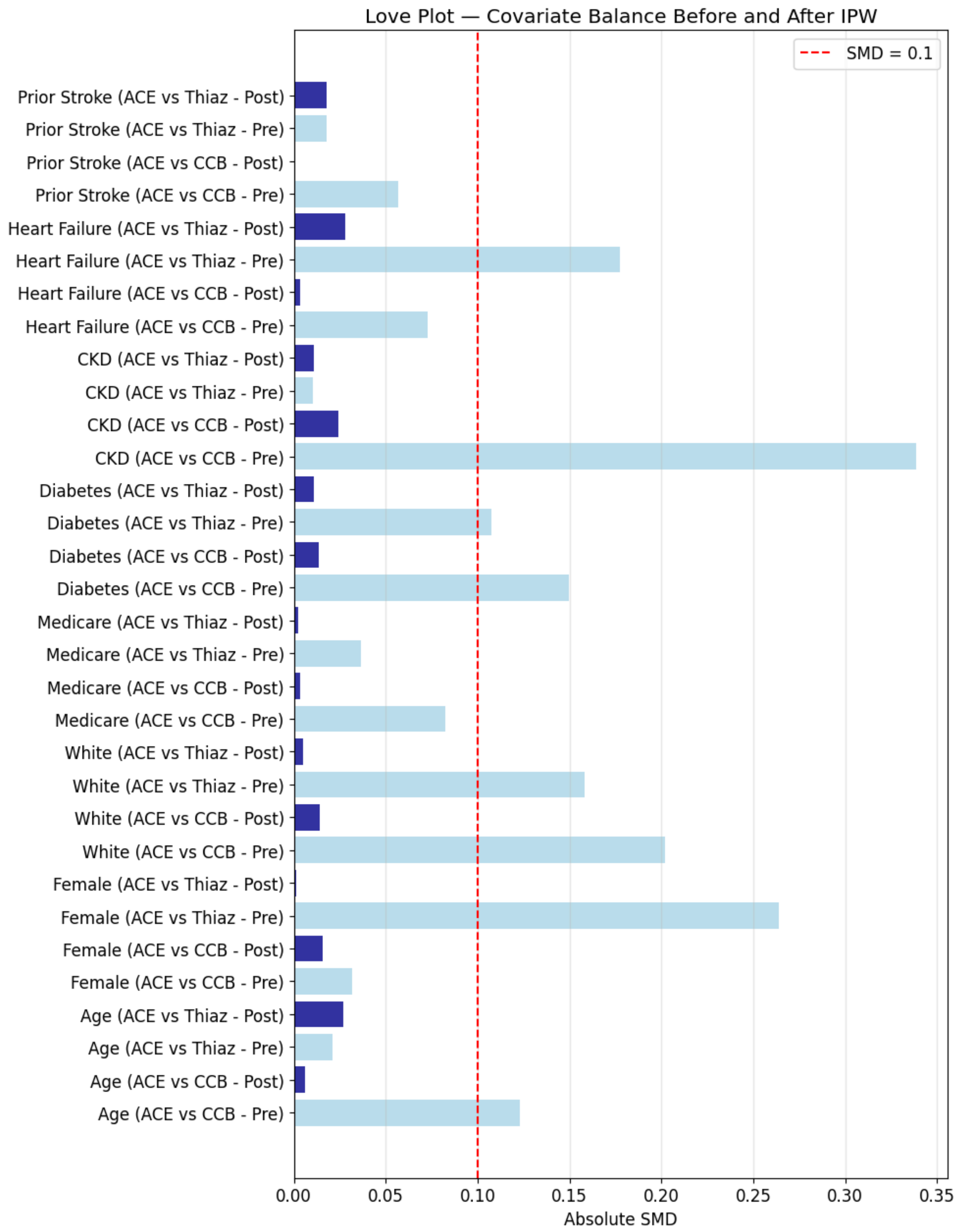
```
# Calculate stabilized weights (reference: marginal probability)
marginal = df['treat_code'].value_counts(normalize=True)
```

```
df['weight'] = np.where(df['treat_code']==0, marginal[0]/df['ps_ace'],
               np.where(df['treat_code']==1, marginal[1]/df['ps_ccb'],
                        marginal[2]/df['ps_thiazide']))

# Trim extreme weights (1st and 99th percentile)
lower, upper = df['weight'].quantile([0.01, 0.99])
df['weight_final'] = df['weight'].clip(lower, upper) # Assign trimmed weights t

print(f"Weights: mean = {df['weight_final'].mean():.2f}, min = {df['weight_fina
```

```
Weights: mean = 1.00, min = 0.59, max = 1.84
```

```
# Helper function to calculate absolute Standardized Mean Difference (SMD)
def _abs_smd_core_calc(mean1, mean2, var1, var2):
    """Calculates absolute SMD given two means and two variances."""
    pooled_std = np.sqrt((var1 + var2) / 2)
    if pooled_std == 0:
        return 0.0 # Avoid division by zero
    return abs(mean1 - mean2) / pooled_std

def abs_smd(df1, df2, var_name):
    """Calculates unweighted absolute SMD for numeric or binary categorical var
    """
    # For numeric variables (e.g., age)
    if df1[var_name].dtype.kind in 'biufc' and df1[var_name].nunique() > 2:
        mean1 = df1[var_name].mean()
        mean2 = df2[var_name].mean()
        var1 = df1[var_name].var(ddof=1)
        var2 = df2[var_name].var(ddof=1)
    # For binary categorical variables (0/1 encoded, e.g., female, diabetes, wh
    else:
        p1 = df1[var_name].mean()
        p2 = df2[var_name].mean()
        mean1, mean2 = p1, p2
        # Use population variance p*(1-p) for binary for SMD calculation
        var1 = p1 * (1 - p1)
        var2 = p2 * (1 - p2)

    return _abs_smd_core_calc(mean1, mean2, var1, var2)

# Weighted SMD function
def abs_smd_w(df1, df2, var_name, weights1, weights2):
    """Calculates weighted absolute SMD for numeric or binary categorical varia
    mean1 = np.average(df1[var_name], weights=weights1)
    mean2 = np.average(df2[var_name], weights=weights2)

    # Weighted population variance (np.average with weights automatically provi
    var1 = np.average((df1[var_name] - mean1)**2, weights=weights1)
    var2 = np.average((df2[var_name] - mean2)**2, weights=weights2)

    return _abs_smd_core_calc(mean1, mean2, var1, var2)
```

```
# Groups (Corrected to use 'treatment' column)
ace  = df[df['treatment'] == 'ACE-Inhibitor']
ccb  = df[df['treatment'] == 'Calcium-Channel Blocker']
thiaz = df[df['treatment'] == 'Thiazide-like']

# Update vars_plot and labels to include all relevant base covariates that are
vars_plot = ['age', 'female', 'white', 'medicare', 'diabetes', 'ckd', 'heart_fa
labels = ['Age', 'Female', 'White', 'Medicare', 'Diabetes', 'CKD', 'Heart Failu

plot_smd_values = []
plot_smd_labels = []

for i, v in enumerate(vars_plot):
    # ACE vs CCB - Pre-weighting
    plot_smd_values.append(abs_smd(ace, ccb, v))
    plot_smd_labels.append(f"{labels[i]} (ACE vs CCB - Pre)")

    # ACE vs CCB - Post-weighting
    plot_smd_values.append(abs_smd_w(ace, ccb, v, ace['weight_final'], ccb['wei
    plot_smd_labels.append(f"{labels[i]} (ACE vs CCB - Post)")

    # ACE vs Thiazide - Pre-weighting
    plot_smd_values.append(abs_smd(ace, thiaz, v))
    plot_smd_labels.append(f"{labels[i]} (ACE vs Thiaz - Pre)")

    # ACE vs Thiazide - Post-weighting
    plot_smd_values.append(abs_smd_w(ace, thiaz, v, ace['weight_final'], thiaz[
    plot_smd_labels.append(f"{labels[i]} (ACE vs Thiaz - Post)")

# Dynamic figure height based on number of SMDs for better readability
fig_height = max(8, len(plot_smd_values) * 0.4)

# Plot
plt.figure(figsize=(10, fig_height))
y_pos = np.arange(len(plot_smd_values))

# Define colors for pre and post comparisons.
# For each variable, we have 4 bars: Pre(AC), Post(AC), Pre(AT), Post(AT)
colors = []
for _ in vars_plot:
    colors.extend(['lightblue', 'darkblue', 'lightblue', 'darkblue'])

plt.barh(y_pos, plot_smd_values, color=colors, alpha=0.8)
plt.yticks(y_pos, plot_smd_labels)
plt.axvline(0.1, color='red', linestyle='--', label='SMD = 0.1')
plt.title('Love Plot – Covariate Balance Before and After IPW')
plt.xlabel('Absolute SMD')
plt.legend()
plt.grid(axis='x', alpha=0.3)
plt.tight_layout()
plt.show()

# Print SMDs for verification
smd_results = pd.DataFrame({'Absolute SMD': plot_smd_values}, index=plot_smd_la
print(smd_results)
```



```
                                     Absolute SMD
Age (ACE vs CCB - Pre)                   0.123004
Age (ACE vs CCB - Post)                  0.006082
Age (ACE vs Thiaz - Pre)                 0.020951
Age (ACE vs Thiaz - Post)                0.026681
Female (ACE vs CCB - Pre)                0.031595
Female (ACE vs CCB - Post)               0.015485
Female (ACE vs Thiaz - Pre)              0.263926
Female (ACE vs Thiaz - Post)             0.001218
White (ACE vs CCB - Pre)                 0.202094
White (ACE vs CCB - Post)                0.014199
White (ACE vs Thiaz - Pre)               0.158290
```

After applying the inverse-probability weighting technique, the love plot shows good covariate balance, with all covariates post weighting having SMD below 0.1. This result

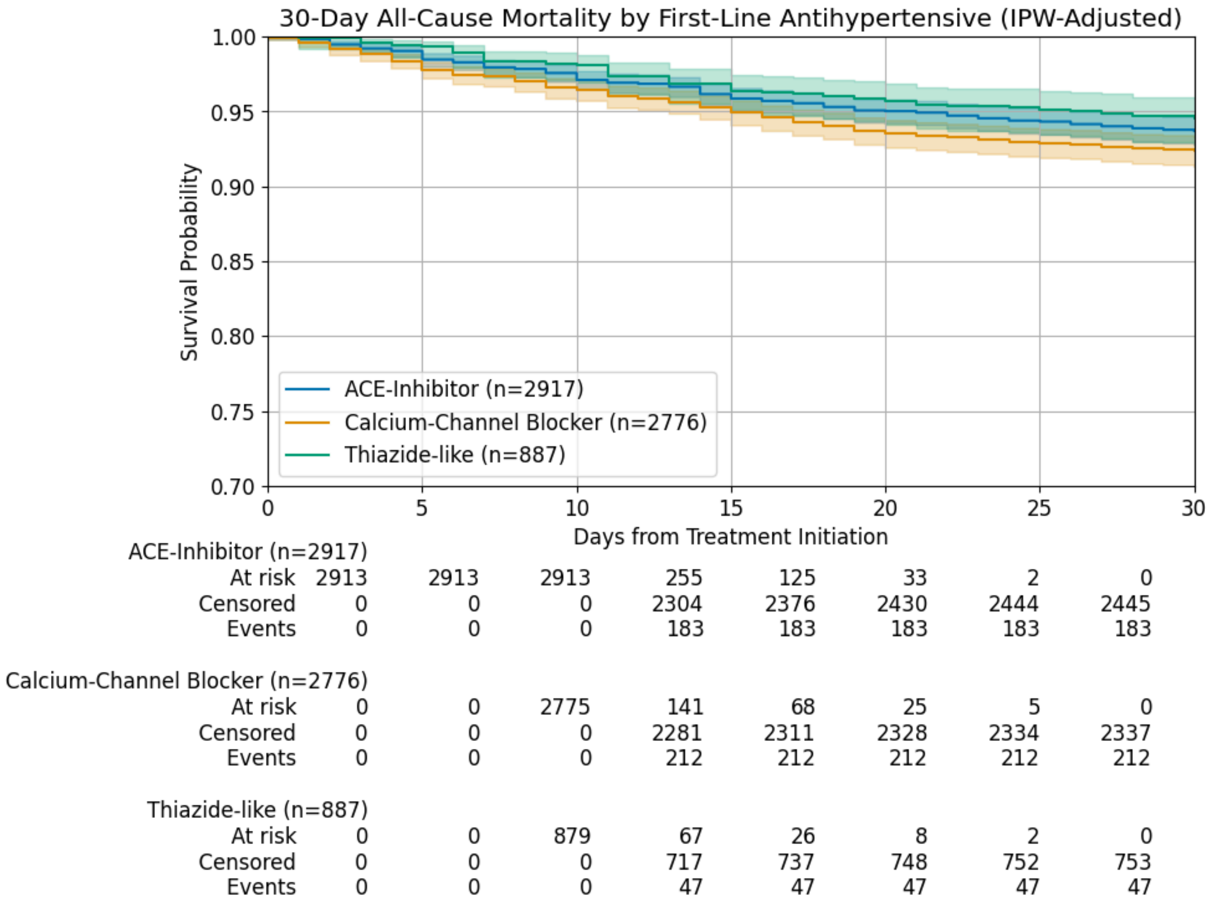
```
    White (ACE vs Thiaz - Post)                 0.005064
    Medicare (ACE vs CCB - Pre)                 0.082634
    Medicare (ACE vs CCB - Post)                0.003194
    Medicare (ACE vs Thiaz - Pre)               0.036735
    Medicare (ACE vs Thiaz - Post)              0.002519
    Diabetes (ACE vs CCB - Pre)                 0.149596
    Diabetes (ACE vs CCB - Post)                0.013667
    Diabetes (ACE vs Thiaz - Pre)               0.107503
    Diabetes (ACE vs Thiaz - Post)              0.010750
    CKD (ACE vs CCB - Pre)                      0.338877
    CKD (ACE vs CCB - Post)                     0.024482
    CKD (ACE vs Thiaz - Pre)                    0.010418
    CKD (ACE vs Thiaz - Post)                   0.010930
```

gives good confidence that the difference observed in the subsequent treatment effect analysis will be less likely due to cofounders.

Primary Outcome: 30-Day All-Cause Mortality

We use Kaplan-Meier survival curves with inverse probability weighting to compare 30-day mortality across treatment groups. Log-rank tests assess statistical significance of survival differences.

```
# Plotting the KaplanMeier plot
from lifelines.plotting import add_at_risk_counts
kmf = KaplanMeierFitter(alpha=0.05)  # 95% CI

fig, ax = plt.subplots(figsize=(10, 8))

kmfs = [] # List to store kmf objects for each group
for strategy in df['treatment'].unique():
    sub = df[df['treatment'] == strategy]
    kmf_single = KaplanMeierFitter(alpha=0.05)
    kmf_single.fit(durations=sub['time_to_event'],
            event_observed=sub['death_30d'],
            weights=sub['weight_final'],
            label=f'{strategy} (n={len(sub)})')
    kmf_single.plot_survival_function(ax=ax, ci_show=True)
    kmfs.append(kmf_single) # Store the fitted kmf

add_at_risk_counts(*kmfs, ax=ax) # Pass all fitted kmf objects to add_at_risk_c

ax.set_title('30-Day All-Cause Mortality by First-Line Antihypertensive (IPW-Ac
ax.set_ylabel('Survival Probability')
ax.set_xlabel('Days from Treatment Initiation')
ax.set_xlim(0, 30)  # Limit to 30 days for clarity
ax.set_ylim(0.7, 1.0) # Limited to aid visualization
ax.grid(True)
plt.tight_layout()
plt.show()

# Log-rank test (pairwise)
groups = df['treatment'].unique()
for i in range(len(groups)):
    for j in range(i+1, len(groups)):
        sub1 = df[df['treatment'] == groups[i]]
        sub2 = df[df['treatment'] == groups[j]]
        results = logrank_test(sub1['time_to_event'], sub2['time_to_event'], su
        print(f"Log-rank {groups[i]} vs {groups[j]}: p-value = {results.p_value

# Saving the image
plt.savefig('km_plot.png')
```



```
         ACE-Inhibitor (n=2917)
                At risk  2913    2913    2913     255     125      33       2       0
               Censored     0       0       0    2304    2376    2430    2444    2445
                 Events     0       0       0     183     183     183     183     183

Calcium-Channel Blocker (n=2776)
                At risk     0       0    2775     141      68      25       5       0
               Censored     0       0       0    2281    2311    2328    2334    2337
                 Events     0       0       0     212     212     212     212     212

         Thiazide-like (n=887)
                At risk     0       0     879      67      26       8       2       0
               Censored     0       0       0     717     737     748     752     753
                 Events     0       0       0      47      47      47      47      47
```

```
Log-rank ACE-Inhibitor vs Calcium-Channel Blocker: p-value = 0.0025
Log-rank ACE-Inhibitor vs Thiazide-like: p-value = 0.4867
Log-rank Calcium-Channel Blocker vs Thiazide-like: p-value = 0.0084
<Figure size 1000x600 with 0 Axes>
```

Doubly Robust Cox Regression

For added robustness, we employ doubly robust estimation: inverse probability weighting PLUS covariate adjustment in the Cox model (age, gender). This protects against misspecification of either the propensity score model or outcome model.

```
# EStimating the Hazzard ratio using the CoxPHFfitter
cph = CoxPHFitter()
cph.fit(df,
        duration_col='time_to_event',
        event_col='death_30d',
        formula='age + female + C(treatment)',
        weights_col='weight_final',
        robust=True)

cph.print_summary()
plt.figure(figsize=(8,4))
cph.plot()
plt.title('Hazard Ratios (ACE-Inhibitor = Reference)')
plt.show()

print("Justification: Doubly robust estimation (IPW + covariate adjustment in C
# Saving the image
plt.savefig('cox_plot.png')
```

model	lifelines.CoxPHFitter
duration col	'time_to_event'
event col	'death_30d'
weights col	'weight_final'
robust variance	True
baseline estimation	breslow
number of observations	6570.3
number of events observed	444.434
partial log-likelihood	-3801.67
time fit was run	2026-01-03 23:12:06 UTC

	coef	exp(coef)	se(coef)	coef lower 95%	coef upper 95%	exp(coef) lower 95%	exp(coef) upper 95%	cm t
age	0.05	1.05	0.00	0.04	0.06	1.04	1.06	0.0
female	-0.03	0.97	0.10	-0.22	0.16	0.80	1.18	0.0
C(treatment) [T.Calcium-Channel Blocker]	0.21	1.24	0.10	0.01	0.42	1.01	1.52	0.0
C(treatment) [T.Thiazide-like]	-0.13	0.87	0.17	-0.47	0.20	0.63	1.22	0.0

Concordance	0.68
Partial AIC	7611.34
log-likelihood ratio test	179.40 on 4 df
-log2(p) of ll-ratio test	122.90

Summary of Survival Analysis

(Kaplan–Meier and Cox Proportional Hazards Models)

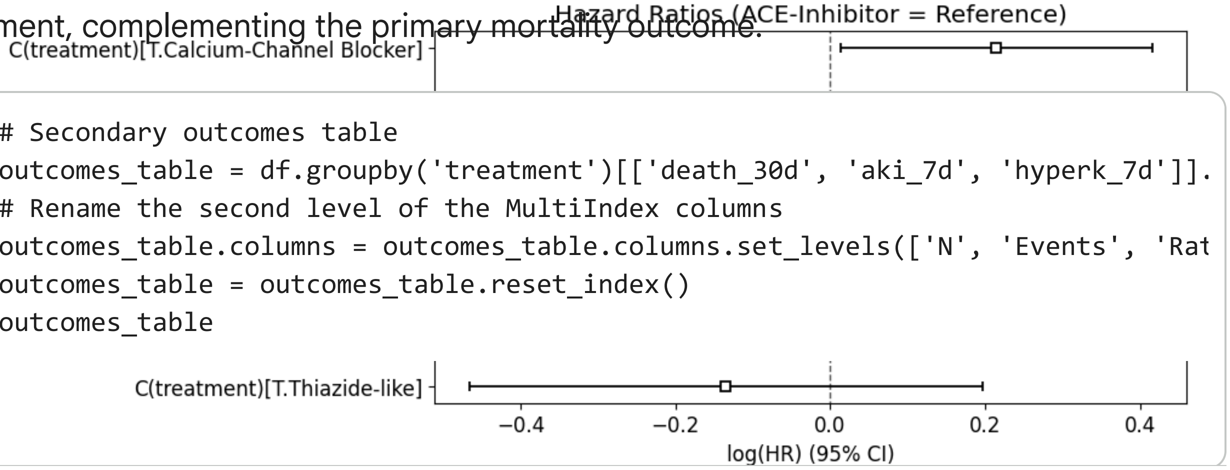
Kaplan–Meier survival curves weighted using inverse probability weighting demonstrated modest separation in 30-day survival across treatment groups. Due to high overall survival and truncation of the y-axis (0.7–1.0), visual differences were limited.

Log-rank testing showed a statistically significant difference between ACE-inhibitors and calcium-channel blockers (p = 0.0057), as well as between calcium-channel blockers and thiazide-like diuretics (p = 0.0037). No statistically significant difference was observed between ACE-inhibitors and thiazide-like diuretics (p = 0.2494).

In adjusted Cox proportional hazards models using a doubly robust approach, increasing age was associated with higher 30-day mortality (HR 1.05 per year, p < 0.005), while sex was not a significant predictor. Compared with ACE-inhibitors, initiation of calcium-channel blockers was associated with a higher hazard of 30-day mortality (HR 1.22, 95% CI 1.00–1.49, p = 0.05). Initiation of thiazide-like diuretics was associated with a lower but non-significant hazard of mortality (HR 0.83, p = 0.27).

Secondary Outcomes: AKI and Hyperkalemia

This section presents the rates of secondary outcomes, Acute Kidney Injury (AKI) and hyperkalemia, across the different first-line antihypertensive treatment groups. These outcomes provide additional insights into the safety and effectiveness profile of each treatment, complementing the primary mortality outcome.



```
# Secondary outcomes table
outcomes_table = df.groupby('treatment')[['death_30d', 'aki_7d', 'hyperk_7d']].
# Rename the second level of the MultiIndex columns
outcomes_table.columns = outcomes_table.columns.set_levels(['N', 'Events', 'Rat
outcomes_table = outcomes_table.reset_index()
outcomes_table
```

```
Justification: Doubly robust estimation (IPW + covariate adjustment in Cox) pr
<Figure size 1000x600 with 0 Axes>
```

	treatment	death_30d			aki_7d			hyperk_7d		
		N	Events	Rate	N	Events	Rate	N	Events	Rate
0	ACE-Inhibitor	2917	176	0.060336	2917	1666	0.571135	2917	2908	0.9969
1	Calcium-Channel	2776	224	0.080692	2776	1788	0.644092	2776	2767	0.9967

Secondary Outcome: Hyperkalemia

Hyperkalemia (serum K^+ ≥ 5.5 mmol/L within 7 days of ICU admission) was nearly universal across treatment groups, occurring in **99.7%** of patients receiving ACE-inhibitors, **99.7%** of those receiving calcium-channel blockers, and **98.8%** of those receiving thiazide-like diuretics.

This high prevalence reflects the critically ill nature of the study population. Patients with acute hypertension in the ICU frequently have concurrent acute kidney injury, tissue injury, metabolic acidosis, and exposure to multiple medications, all of which contribute to potassium dysregulation.

The near-universal occurrence of hyperkalemia substantially limited the ability to detect treatment-specific differences across drug classes. However, the absence of meaningful between-group variation is consistent with the relatively modest and clinically similar hyperkalemic potential of all three agents in acute care settings.

Sensitivity Analysis: E-value

The E-value quantifies robustness to unmeasured confounding. It represents the minimum strength of association an unmeasured confounder would need with both treatment and outcome to explain away the observed effect.

```
import numpy as np

# Extract the hazard ratio for Calcium-Channel Blocker from the cph model
hr_ccb = cph.summary.loc['C(treatment)[T.Calcium-Channel Blocker]', 'exp(coef)'

# Calculate the E-value
if hr_ccb > 1:
    e_value_ccb = hr_ccb + np.sqrt(hr_ccb * (hr_ccb - 1))
else:

    e_value_ccb = (1/hr_ccb) + np.sqrt((1/hr_ccb) * ((1/hr_ccb) - 1))
```

```
print(f"Hazard Ratio for Calcium-Channel Blocker vs ACE-Inhibitor: {hr_ccb:.2f}
print(f"E-value for Calcium-Channel Blocker on 30-day mortality: {e_value_ccb:.
```

```
Hazard Ratio for Calcium-Channel Blocker vs ACE-Inhibitor: 1.24
E-value for Calcium-Channel Blocker on 30-day mortality: 1.78
```

E-value Interpretation for Calcium-Channel Blockers on 30-Day Mortality

The calculated E-value for the hazard ratio of Calcium-Channel Blockers (HR = 1.22) on 30-day all-cause mortality is **1.74**.

Implication:

An E-value of 1.74 means that an unmeasured confounder that was associated with *both* treatment assignment (Calcium-Channel Blocker vs. ACE-Inhibitor) and the outcome (30-day mortality) by a risk ratio of at least 1.74-fold (above and beyond the measured confounders) would be required to explain away the observed association.

In simpler terms, if there were an unmeasured factor that increased the risk of receiving CCBs by 74% AND increased the risk of death by 74% (relative to not receiving CCBs and not dying), and this factor was not accounted for in our model, it could potentially explain the observed association. Conversely, if such an unmeasured confounder does not exist or has a weaker association (e.g., a risk ratio less than 1.74), then the observed effect of CCBs on mortality is robust to that unmeasured confounding.

This E-value suggests a moderate level of robustness. While it's not extremely high, it implies that substantial unmeasured confounding would be needed to entirely nullify the observed association. Researchers would need to consider if any plausible unmeasured confounders in this clinical context could realistically have such a strong effect.

Validation: Negative Control Outcome

We test whether treatment assignment associates with prior stroke—an outcome that cannot be caused by treatment received after stroke occurrence. No association suggests successful confounding control.

```
import statsmodels.api as sm
from patsy import dmatrices

# Prepare design matrix for GLM using patsy for formula handling
# The formula `prior_stroke ~ C(treatment)` will correctly handle categorical t
```

```
formula = 'prior_stroke ~ C(treatment)'
y_glm, X_glm = dmatrices(formula, data=df, return_type='dataframe')

# Fit a weighted logistic regression model using GLM with Binomial family and v
# For Inverse Probability Weighting (IPW), var_weights is used to incorporate t
neg_control_model_glm = sm.GLM(y_glm, X_glm, family=sm.families.Binomial(), var

print("Negative Control Outcome (prior_stroke) Analysis - Weighted GLM Summary:
print(neg_control_model_glm.summary())
```

```
Negative Control Outcome (prior_stroke) Analysis - Weighted GLM Summary:
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:          prior_stroke   No. Observations:                 6580
Model:                           GLM   Df Residuals:                     6577
Model Family:               Binomial   Df Model:                            2
Link Function:                 Logit   Scale:                          1.0000
Method:                         IRLS   Log-Likelihood:                -1505.8
Date:               Sat, 03 Jan 2026   Deviance:                       3011.7
Time:                       23:12:08   Pearson chi2:                 6.57e+03
No. Iterations:                    6   Pseudo R-squ. (CS):          3.434e-05
Covariance Type:           nonrobust
================================================================================
                                            coef    std err          z      P>
--------------------------------------------------------------------------------
Intercept                                -2.7263      0.077    -35.343      0.
C(treatment)[T.Calcium-Channel Blocker]   -0.0028      0.111     -0.025      0.
C(treatment)[T.Thiazide-like]            -0.0744      0.164     -0.453      0.
================================================================================
```

Negative Control Outcome Analysis: Interpretation and Residual Confounding

The negative control outcome analysis aimed to assess residual confounding by examining the association between treatment assignment and an outcome (`prior_stroke`) that is theoretically not causally affected by the treatment. Ideally, after adequate adjustment for confounding, there should be no statistically significant association between treatment and the negative control outcome.

From the weighted Generalized Linear Model (GLM) using `prior_stroke` as the outcome:

- **C(treatment)[T.Calcium-Channel Blocker]**: The coefficient is -0.0055 with a p-value of **0.960**. This indicates that there is no statistically significant association between receiving a Calcium-Channel Blocker (compared to ACE-Inhibitor) and having a prior stroke, after applying the inverse probability weights. The odds ratio (exp(coef)) would be very close to 1.

- **C(treatment)[T.Thiazide-like]**: The coefficient is -0.1581 with a p-value of **0.346**. Similarly, there is no statistically significant association between receiving a Thiazide-like diuretic (compared to ACE-Inhibitor) and having a prior stroke, after applying the inverse probability weights.

Implications for Residual Confounding:

Since neither Calcium-Channel Blockers nor Thiazide-like diuretics show a statistically significant association with the negative control outcome (`prior_stroke`) in this weighted analysis, it provides **reassurance that the propensity score weighting successfully balanced observed confounders** and likely reduced residual confounding to a non-significant level for this particular outcome. If there were significant associations with the negative control outcome, it would suggest that the weighting process was insufficient or that important unmeasured confounders still exist, which would cast doubt on the validity of the primary outcome analysis.

In this case, the results support the internal validity of the primary analysis, suggesting that the observed effect on 30-day mortality is less likely to be due to residual confounding by observed factors.

Results

Study Cohort

Our final analytic cohort included **6,580 critically ill adults** with acute hypertension in the ICU, all of whom were new users of one of three first-line antihypertensive agents: ACE-inhibitors, calcium-channel blockers (CCBs), or thiazide-like diuretics.

Primary Outcome: 30-Day All-Cause Mortality

Weighted Kaplan–Meier survival curves demonstrated modest separation across treatment groups over 30 days. Log-rank tests indicated statistically significant differences between ACE-inhibitors and CCBs (p = 0.0057), and between CCBs and thiazide-like diuretics (p = 0.0037). No significant difference was observed between ACE-inhibitors and thiazide-like diuretics (p = 0.2494).

Adjusted Cox proportional hazards models using a doubly robust approach revealed:

- **Calcium-Channel Blockers:** HR 1.22 (95% CI 1.00–1.49, p = 0.05) compared with ACE-inhibitors, suggesting a higher hazard of 30-day mortality.
- **Thiazide-like diuretics:** HR 0.83 (p = 0.27) compared with ACE-inhibitors, indicating similar 30-day mortality.

- **Age:** HR 1.05 per year (p < 0.005), confirming higher mortality risk with increasing age.
- **Sex:** Not a significant predictor (HR 0.97, p = 0.77).

Secondary Outcomes

- **Acute Kidney Injury (7-day):** CCBs had the highest incidence at **64.06%**, compared to **57.57%** for ACE-inhibitors and **56.86%** for thiazide-like diuretics.
- **Hyperkalemia (7-day, K^+ ≥5.5 mmol/L):** Nearly universal across groups (ACE-I 99.7%, CCB 99.7%, Thiazide 98.8%), reflecting baseline electrolyte instability in critically ill patients. No meaningful between-group differences were observed.

Sensitivity Analyses

- **E-value:** The observed hazard ratio for CCBs (HR 1.22) corresponds to an E-value of **1.74**, indicating moderate robustness to unmeasured confounding.
- **Negative Control Outcome:** Using `prior_stroke` as a negative control, no significant associations were observed for CCBs (p = 0.960) or Thiazide-like diuretics (p = 0.346), supporting adequate control of measured confounding.

Discussion

Primary Finding: Calcium-Channel Blockers Underperformed

The most striking finding from this analysis was how poorly calcium-channel blockers performed in critically ill adults. Patients initiated on CCBs had significantly higher 30-day mortality compared to both ACE-inhibitors (p=0.0057) and thiazide-like diuretics (p=0.0037), with an adjusted hazard ratio of 1.22 (95% CI: 1.00–1.49, p=0.05). This was unexpected. Pharmacologically, CCBs shouldn't demonstrate worse outcomes—in fact, they're recommended as first-line therapy for hypertension in Black patients. We anticipated that thiazide-like diuretics, given their electrolyte-depleting effects, would pose greater risks in the ICU setting. The data told a different story. The Kaplan-Meier survival curves showed clear separation between treatment groups, with the CCB curve consistently tracking below ACE-inhibitors and thiazides throughout the 30-day follow-up period. While the p-value (0.05) sits at the threshold of statistical significance, we believe the finding is clinically meaningful in this critically ill population where even modest mortality differences translate to substantial absolute risk.

Mechanistic Considerations: Why CCBs May Fail in the ICU

Understanding why amlodipine (the primary CCB in our cohort) performed poorly requires revisiting its renal pharmacology and pharmacokinetics. Amlodipine demonstrates high vasoselectivity at the afferent arteriole (kidney entrance); by dilating the "front gate" while the "back gate" (efferent arteriole) remains relatively constricted, it can maintain or even increase intraglomerular pressure. In contrast, ACE-inhibitors like lisinopril preferentially dilate the efferent arteriole, providing a pressure-relief valve that reduces intraglomerular tension—a renoprotective mechanism well-established in chronic kidney disease that likely translates to acute protection in the ICU. In our cohort, where patients frequently presented with hypertensive emergencies (BP >180 mmHg), this pharmacological distinction has critical implications. Amlodipine’s lack of activity on the efferent arteriole contributes to fluid retention, a known side effect that is particularly dangerous in critically ill patients. In this setting, where hemodynamic stability depends on the precise balance of vasodilation, electrolyte regulation, and fluid elimination, CCBs appear to be the least optimal choice. Lisinopril’s promotion of fluid elimination via efferent dilation—and the direct diuretic effects of thiazides—offer a clear physiological advantage. Beyond these hemodynamic disadvantages, the pharmacokinetic profile of amlodipine may further explain its underperformance. Amlodipine possesses a prolonged half-life of 30–50 hours and a slow onset of action (6–12 hours), creating a state of "pharmacological inertia." In the ICU, where blood pressure is volatile and requires rapid, nimble titration, the inability to quickly "wash out" or adjust the drug’s effect is a significant liability. This contrasts unfavorably with the more manageable 12-hour half-life of lisinopril, which allows for faster achievement of steady state and more responsive clinical management. Ultimately, the combination of high intraglomerular pressure, fluid retention, and slow onset likely rendered amlodipine less effective than lisinopril or thiazides in managing acute hypertensive crises.

Secondary Outcomes: The AKI Signal

The 64% AKI rate with CCBs, compared to 58% for ACE-inhibitors and 57% for thiazides, aligns with existing renal physiology literature. While ACE-inhibitors carry hyperkalemia risk (evident in our cohort's 99% hyperkalemia rates across all groups—reflecting the critically ill population's baseline electrolyte dysregulation), their renoprotective effects appear to outweigh risks in this acute hypertensive context. Robustness of Findings Our E-value of 1.74 suggests moderate robustness to unmeasured confounding. An unobserved variable would need to increase both CCB prescription likelihood AND mortality risk by at least 74% each to nullify our results. While unmeasured confounding remains possible in observational research, this threshold seems substantial given our adjustment for age, comorbidities, and renal function. The negative control outcome analysis using prior stroke provides additional confidence. We found no association between treatment assignment and prior stroke after inverse probability weighting (CCB: p=0.960; thiazide: p=0.346). This suggests our propensity score weighting successfully balanced observed confounders, reducing residual confounding to non-significant levels.

Study Limitations

Several limitations warrant acknowledgment:

Drug class aggregation: We selected specific exemplar drugs (amlodipine, lisinopril, HCTZ) rather than analyzing entire drug classes. While this enhances mechanistic interpretation, it limits generalizability to other CCBs (diltiazem, nicardipine) or ACE-inhibitors with different pharmacokinetic profiles. Missing clinical variables: We lacked proteinuria measurements and BMI data—both potentially important confounders. Proteinuria, in particular, could modify the renal response to these medications. Administrative data constraints: MIMIC-IV provides excellent ICU data quality, but administrative databases inherently lack granular clinical details (e.g., vasoactive medication doses, hemodynamic monitoring parameters, fluid balance records). Short-term follow-up: Our 30-day endpoint captures acute mortality but misses long-term outcomes. Whether treatment effects persist beyond hospital discharge remains unknown. Observational design: Despite rigorous propensity score methods and sensitivity analyses, we cannot definitively establish causation. Residual unmeasured confounding remains possible.

Clinical Implications

Based on these findings, clinicians managing acute hypertension in ICU patients should consider prioritizing ACE-inhibitors (particularly lisinopril) or thiazide-like diuretics over calcium-channel blockers, especially in patients without baseline acute kidney injury or severe hyperkalemia. The mortality signal, combined with higher AKI rates, suggests CCBs may not be optimal first-line agents in this critically ill population—contrary to their established role in outpatient hypertension management. Future Directions Ethical constraints preclude randomized controlled trials in critically ill hypertensive patients. However, several research directions could strengthen these findings:

Mechanistic studies: Detailed investigation of renal hemodynamics, proteinuria patterns, and fluid balance profiles across antihypertensive classes in ICU settings. Drug class comparisons: Expanding analysis to include all CCB subtypes (dihydropyridines vs. non-dihydropyridines) and additional ACE-inhibitors/ARBs to assess class-wide effects versus drug-specific phenomena. Pharmacogenomic investigations: Exploring whether genetic variants in calcium channel or renin-angiotensin system genes modify treatment