Commodity and Energy ETF Performance Analysis (2015–2025)

As a **Materials Science and Metallurgy Engineer**, understanding how commodities behave in financial markets is essential for connecting **material supply dynamics** with **macroeconomic investment trends**.

This notebook investigates the financial performance of key commodity and energy ETFs, linking their market behavior to real-world developments in **metals**, **battery materials**, and **traditional energy sectors**.

Over the past decade, rapid industrialization, decarbonization policies, and the rise of electric mobility have reshaped commodity demand.
Metals like **lithium (LIT ETF)**, **silver (SLV)**, and **industrial metals (XME)** are now central to the global transition from fossil fuels toward renewable and battery-based systems.
Meanwhile, **traditional energy (XLE)** and **broad market benchmarks (SPY)** serve as controls for evaluating diversification and performance trade-offs.

Objectives

1. **Analyze** the historical performance of metals and energy ETFs from 2015–2025.
2. **Compare** returns, volatility, and growth patterns across commodity categories.
3. **Identify** potential portfolio allocation or hedging strategies for energy-transition-focused investors.
4. **Interpret** results from a materials science perspective — connecting ETF performance to real-world **supply/demand cycles**, **energy policies**, and **technological shifts**.

In [2]:
```
import pandas as pd
import numpy as np
import yfinance as yf
import matplotlib.pyplot as plt

pd.options.display.float_format = "{:.6f}".format
```

In [40]:
```
tickers = ["GLD", "SLV", "LIT", "XME", "XLE", "SPY"]
start_date = "2015-01-01"

data = yf.download(tickers, start=start_date, auto_adjust=True)
data = data["Close"].dropna()
data.head()
```

```
[*******************100%***********************]  6 of 6 completed
```

Out[40]:

Ticker Date	GLD	LIT	SLV	SPY	XLE	XME
2015-01-02	114.080002	19.526278	15.110000	171.093704	51.868366	26.559984
2015-01-05	115.800003	19.109713	15.500000	168.003815	49.722694	25.579153
2015-01-06	117.120003	19.109713	15.830000	166.421371	48.992241	25.226400
2015-01-07	116.430000	19.109713	15.850000	168.495148	49.096569	25.338249
2015-01-08	115.940002	19.335354	15.640000	171.485092	50.198769	25.579153

Data Loading and Setup

To begin, we import the core Python libraries required for **data collection**, **manipulation**, and **visualization**.

We use `yfinance` to retrieve historical price data for selected ETFs representing:

- **Precious metals** → Gold (GLD), Silver (SLV)
- **Battery metals** → Lithium (LIT)
- **Industrial metals** → XME
- **Energy sector** → XLE
- **Market benchmark** → S&P 500 (SPY)

All price data are adjusted for splits and dividends to ensure accurate return calculations.

In [41]:

```
ret_d = data.pct_change().dropna()
ret_l = np.log(data).diff().dropna()

ret_d.head()
```

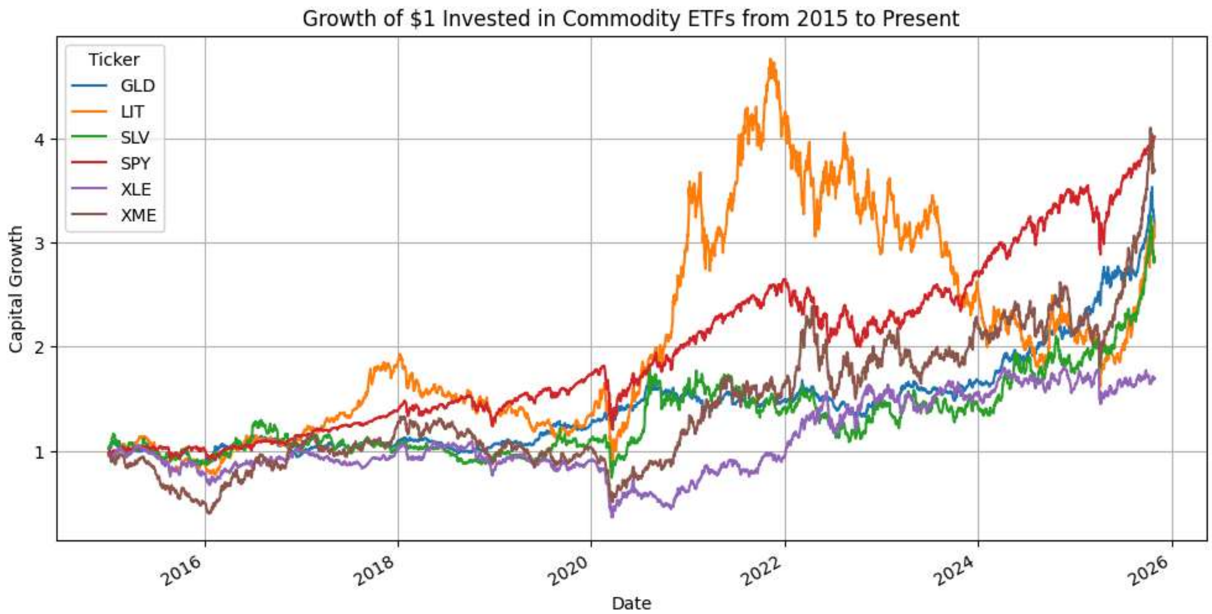
Out[41]:

Ticker Date	GLD	LIT	SLV	SPY	XLE	XME
2015-01-05	0.015077	-0.021334	0.025811	-0.018060	-0.041368	-0.036929
2015-01-06	0.011399	0.000000	0.021290	-0.009419	-0.014691	-0.013791
2015-01-07	-0.005891	0.000000	0.001263	0.012461	0.002129	0.004434
2015-01-08	-0.004209	0.011808	-0.013249	0.017745	0.022450	0.009508
2015-01-09	0.011385	-0.000898	0.008312	-0.008013	-0.007925	-0.002354

In [42]:

```
#  Preliminary Visual Analysis: Cumulative Growth Curve

# Using discrete returns (ret_d) and calculate the growth of $1 invested in each ET
# The .cumprod() method calculates the cumulative product, effectively compounding
```

```
(1 + ret_d).cumprod().plot(figsize=(12, 6),
                           title="Growth of $1 Invested in Commodity ETFs from 2015

plt.ylabel("Capital Growth")
plt.xlabel("Date")
plt.grid(True)
plt.show()
```



Cumulative Growth of $1 Invested

This visualization tracks how **$1 invested in each ETF since 2015** would have grown over time.

The chart highlights:

- The **outperformance** of lithium (LIT) during the electric mobility boom (2020–2022).
- The **steady growth** of gold (GLD) and silver (SLV) as traditional hedging assets.
- The **volatility** of industrial metals (XME) and energy (XLE) during the COVID-19 recovery phase.
- **SPY** serves as a benchmark representing general market performance.

This step provides a macro-level view of how commodity-linked sectors reacted to global industrial and policy trends.

In [45]:
```
TRADING_DAYS = 252

mean_annual = ret_d.mean() * TRADING_DAYS
vol_annual = ret_d.std() * np.sqrt(TRADING_DAYS)

# Sharpe (rf=0)
sharpe = mean_annual / vol_annual

# Sortino (rf=0)
```

```
downside = ret_d.clip(upper=0)
downside_std = downside.std() * np.sqrt(TRADING_DAYS)
sortino = mean_annual / downside_std

metrics = pd.DataFrame({
    'mean_annual': mean_annual,
    'vol_annual': vol_annual,
    'sharpe': sharpe,
    'sortino': sortino
}).sort_values('sharpe', ascending=False)

metrics
```

Out[45]:

Ticker	mean_annual	vol_annual	sharpe	sortino
SPY	0.144786	0.178689	0.810267	1.258745
GLD	0.117953	0.146739	0.803830	1.343077
XME	0.176748	0.333442	0.530069	0.883827
LIT	0.151152	0.297009	0.508914	0.852075
SLV	0.133126	0.267232	0.498168	0.817893
XLE	0.092832	0.295142	0.314532	0.498434

Performance Metrics — Risk-Adjusted Evaluation

The following metrics evaluate the risk-return profile of each ETF:

Metric	Description
Annualized Return	Average yearly return assuming compounding.
Annualized Volatility	Measures total risk (standard deviation of returns).
Sharpe Ratio	Risk-adjusted return, accounting for total volatility.
Sortino Ratio	Similar to Sharpe but penalizes only downside risk.

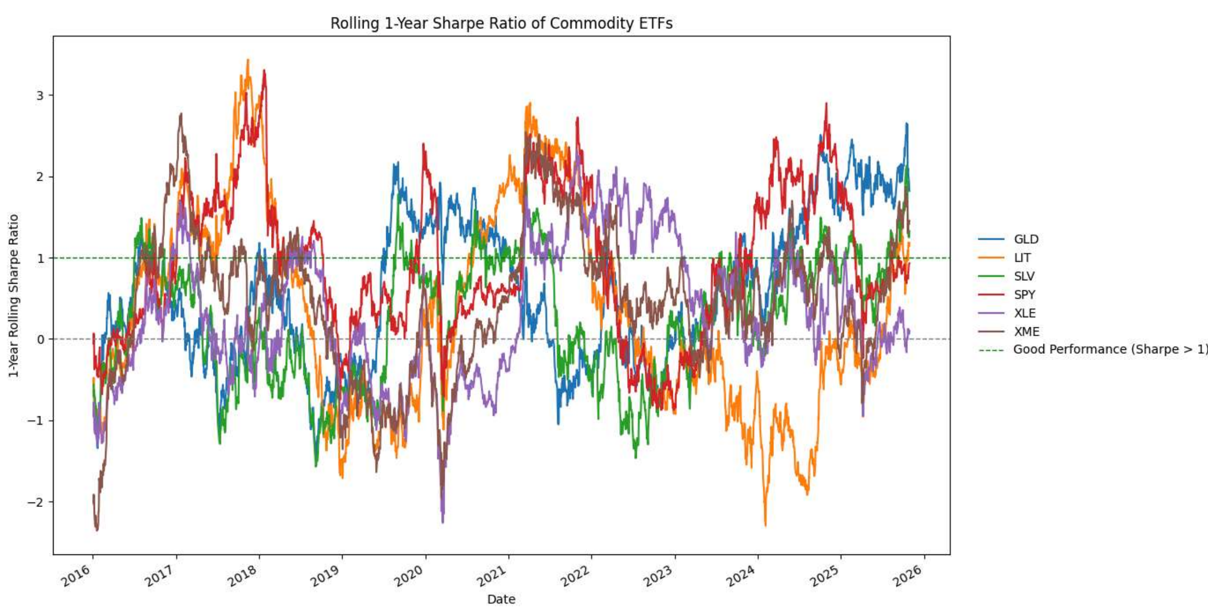
From the results:

- **SPY** (broad market benchmark) leads in overall risk-adjusted performance, with a **Sharpe ratio ≈ 0.81**, indicating stable long-term equity returns.
- **GLD** (Gold ETF) shows the **highest Sortino ratio**, suggesting strong downside protection and stable performance — consistent with gold's role as a **safe-haven metal**.
- **XME (Industrial Metals)** exhibits the **highest annual return (≈ 17.6%)**, but at the cost of higher volatility (≈ 33%), reflecting the cyclical nature of metal demand.

- **LIT (Lithium ETF)** also shows strong average returns, tied to the **EV-battery materials boom** post-2020, though with moderate volatility.
- **Energy (XLE)** is the weakest performer, with both low return and high volatility — mirroring oil market instability through 2020 and energy transition headwinds.

Engineering Insight 🧪:
From a materials science perspective, this demonstrates how **market volatility mirrors physical supply-demand shocks** in commodities — such as mine disruptions, energy transitions, or battery metal shortages.

In [46]:
```
# Rolling 1-Year Sharpe Ratio ---

periods_per_year = 252
riskfree_rate_annual = 0.02


rolling_vol = ret_d.rolling(window=periods_per_year).std() * np.sqrt(periods_per_ye
rolling_ret = ret_d.rolling(window=periods_per_year).mean() * periods_per_year


rolling_sharpe = (rolling_ret - riskfree_rate_annual) / rolling_vol

# Plot outcome
ax = rolling_sharpe.plot(figsize=(14,7), title="Rolling 1-Year Sharpe Ratio of Comm
ax.axhline(0, color='grey', linestyle='--', linewidth=1)
ax.axhline(1, color='green', linestyle='--', linewidth=1, label='Good Performance (
ax.set_ylabel("1-Year Rolling Sharpe Ratio")
ax.set_xlabel("Date")

ax.legend(loc='center left', bbox_to_anchor=(1.02, 0.5), frameon=False)
plt.tight_layout()
```



Rolling 1-Year Sharpe Ratio — Time-Varying Risk-Reward Dynamics

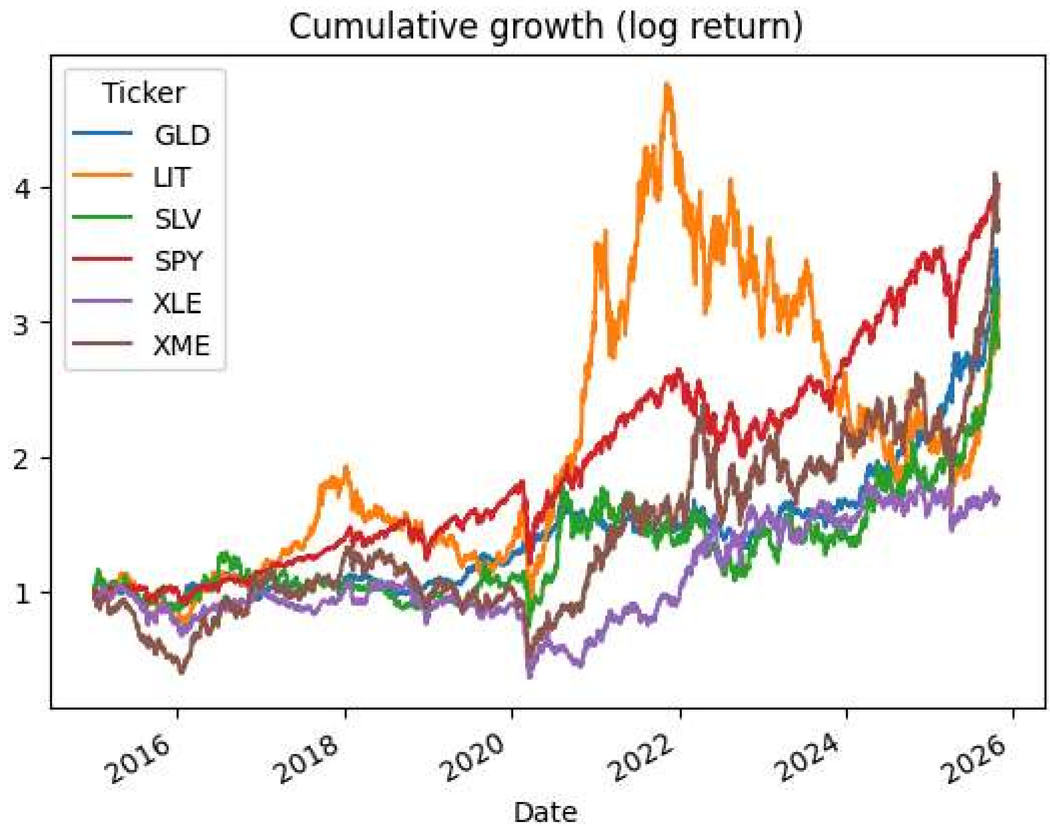
A 1-year rolling Sharpe ratio captures how each ETF's risk-adjusted performance evolves through time.

Interpretation:

- **2017–2020:**
  - Lithium (LIT) and metals (XME) outperform significantly with Sharpe > 1.5, corresponding to industrial demand recovery and early EV adoption.
- **2020–2021:**
  - Extreme fluctuations coincide with the **COVID-19 shock** and subsequent **green energy rally**.
- **2022–2024:**
  - Sharp declines in LIT's Sharpe ratio reflect **commodity overcapacity and demand normalization**.
  - Gold (GLD) maintains stable Sharpe ratios > 1 during market stress periods — reinforcing its defensive properties.
- **Energy (XLE)** shows cyclical peaks and troughs linked to oil supply shocks and OPEC policy swings.

Metallurgical Connection 🛰️:
Volatility in Sharpe ratios reflects **real-world disruptions in mining, refining, and energy logistics**.
Sharp downturns correspond to **supply chain bottlenecks, decarbonization shifts, and raw material scarcity**.

In [47]:
```
cum_log = ret_l.cumsum().apply(np.exp)
ax = cum_log.plot(title="Cumulative growth (log return)")
plt.show()
```



Cumulative Growth — Log Return Perspective

Log cumulative growth smooths exponential effects and helps visualize compounding returns across ETFs.

Key Observations:

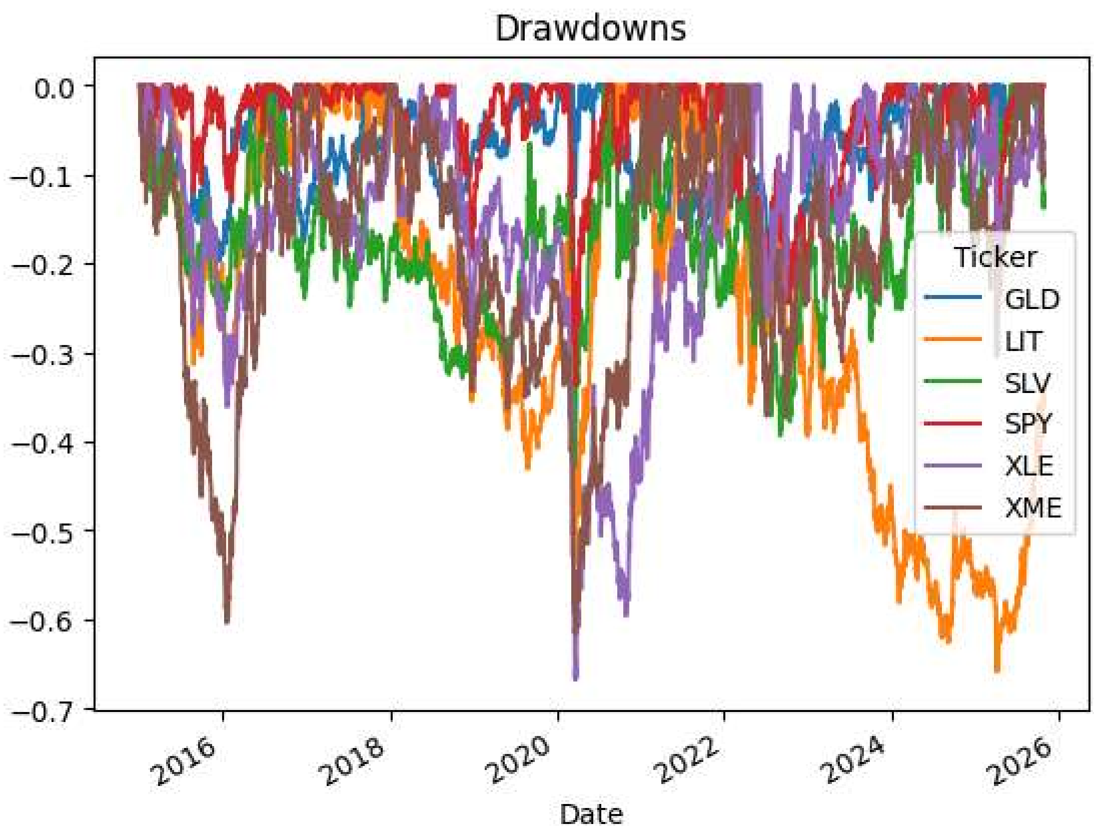
- **LIT (Lithium)** dominates overall growth between 2020–2022, driven by EV demand and lithium price spikes.
- **SPY** maintains steady growth — acting as a reference for long-term portfolio balance.
- **XME** and **GLD** rise notably post-2020, showing resilience in industrial metals and safe-haven assets.
- **XLE (Energy)** remains lagging until the **post-pandemic oil price rebound** in 2022–2023.

Interpretation:
Commodities linked to **clean energy and battery materials** show accelerated growth, supporting the **decarbonization megatrend**.
This phase aligns with global industrial policies emphasizing **critical mineral extraction and processing capacity**.

In [48]:
```
def calculate_drawdown(series: pd.Series):
    roll_max = series.cummax()
    drawdown = (series/roll_max) - 1.0
    return drawdown
```

```
drawdowns = data.apply(calculate_drawdown)
max_dd = drawdowns.min()

print("Max Drawdown:")
print(max_dd)

drawdowns.plot(title="Drawdowns")
plt.show()
```

```
Max Drawdown:
Ticker
GLD   -0.220022
LIT   -0.659128
SLV   -0.428061
SPY   -0.337173
XLE   -0.668130
XME   -0.616914
dtype: float64
```



Drawdowns — Measuring Downside Exposure

A drawdown measures the percentage drop from a portfolio’s previous peak. It helps identify the depth and frequency of losses.

Results Summary:

ETF	Max Drawdown	Interpretation
GLD	-22%	Gold's defensive nature limits losses during crises.
LIT	-65%	Severe cyclical drawdowns from lithium oversupply and price crashes.
SLV	-42%	Moderate drawdowns due to industrial + monetary dual demand.
SPY	-34%	Typical equity-level drawdown during bear markets.
XLE	-67%	Reflects oil price collapses and energy policy volatility.
XME	-61%	High exposure to economic cycles in metals demand.

Engineering Viewpoint ⚙️:
Materials supply chains amplify volatility — for example, lithium and steel (in XME) suffer deep drawdowns when **extraction costs** rise or **industrial demand** weakens.
Gold, however, stabilizes portfolios due to **counter-cyclical investor demand**.

In [49]:

```
from scipy.stats import shapiro

WINDOW = 5000  # last obs for Shapiro-Wilk

def shapiro_last_window(s, n=WINDOW):
    s_ = s.dropna().tail(n)
    if len(s_) < 3:
        return float("nan")
    return shapiro(s_).pvalue

dist_stats = pd.DataFrame({
    "skew": ret_d.skew(),
    "kurtosis": ret_d.kurtosis(),
    "shapiro_p": ret_d.apply(shapiro_last_window)
}).sort_values("shapiro_p")

dist_stats
```

Out[49]:

Ticker	skew	kurtosis	shapiro_p
SPY	-0.313079	14.132080	0.000000
XLE	-0.424698	13.218580	0.000000
SLV	-0.184778	5.602591	0.000000
LIT	0.055784	4.947264	0.000000
XME	-0.128490	3.630908	0.000000
GLD	-0.153646	3.171798	0.000000

Statistical Properties of Daily Returns — Skew, Kurtosis, and Normality

To test the statistical nature of returns:

- **Skew** → asymmetry in returns distribution (positive = right tail, negative = left tail)
- **Kurtosis** → measures fat tails (higher = more extreme outliers)
- **Shapiro–Wilk Test** → checks for normality ($p < 0.05$ → not normal)

Findings:

- All ETFs show **non-normal distributions** ($p \approx 0.00$), typical for financial returns.
- **SPY** and **XLE** have high kurtosis (>13), indicating **frequent extreme price movements**.
- **LIT** and **XME** exhibit slight positive skew — occasional explosive rallies during commodity booms.
- **GLD** and **SLV** maintain moderate tails and near-zero skew, reflecting balanced trading dynamics.

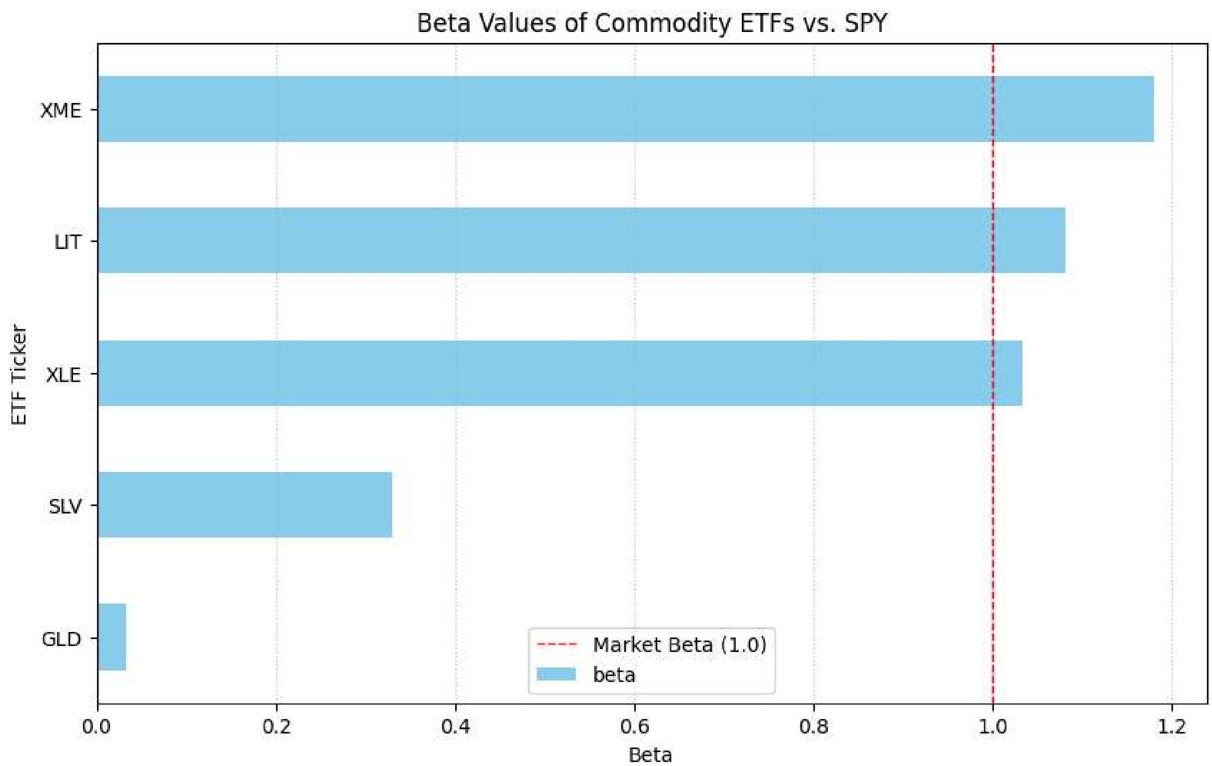
Interpretation:
Metal and energy ETFs exhibit **leptokurtic (fat-tailed)** distributions — consistent with **real-world supply shocks** such as mine closures, geopolitical risks, or oil embargoes.
This statistical asymmetry underscores that **commodity returns are inherently non-Gaussian**, shaped by both physical production constraints and speculative demand surges.

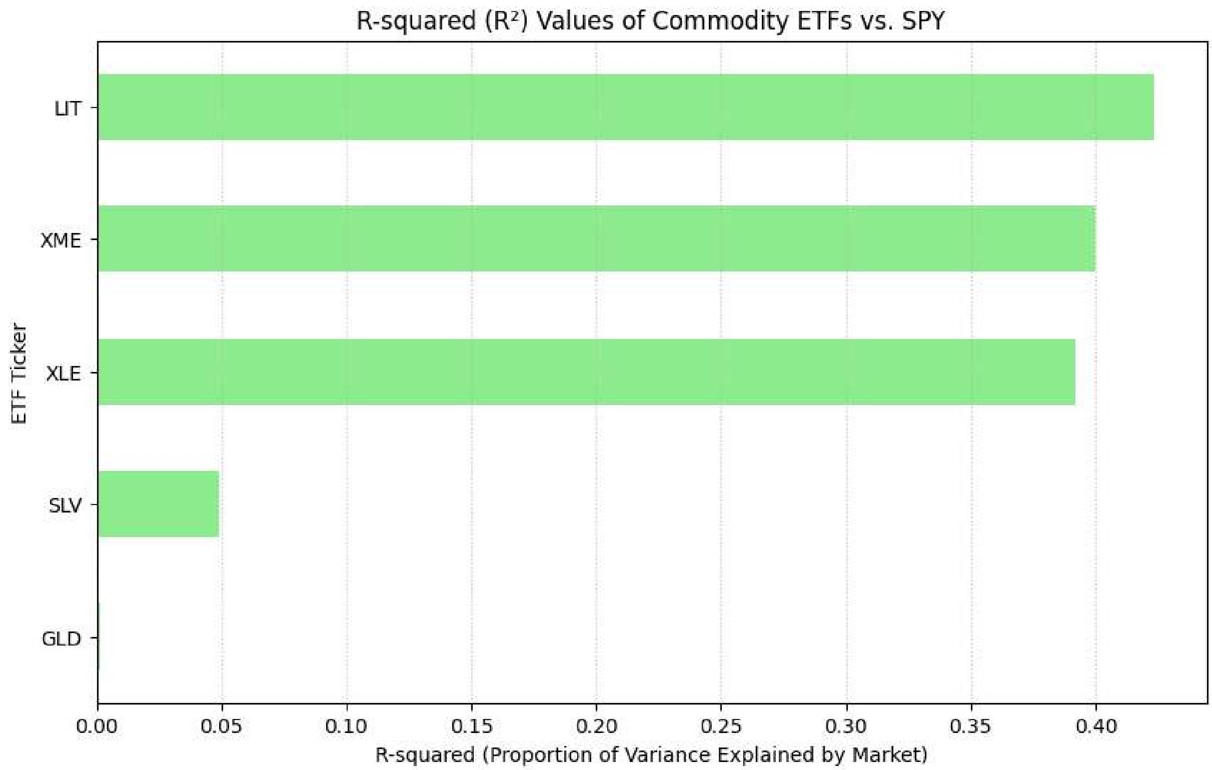
In [50]:
```python
# CAPM vs S&P 500 (Market Benchmark)

import statsmodels.api as sm
import matplotlib.pyplot as plt

#  Setup: Define the market benchmark and the assets to be analyzed
market_ticker = 'SPY'

asset_tickers = [ticker for ticker in ret_d.columns if ticker != market_ticker]

# A list to store the results for each asset's regression
capm_results_list = []

print(f"Calculating CAPM for each asset vs. {market_ticker}...")

#  Loop to run the regression for each asset
for ticker in asset_tickers:
    # Prepare the data: align dates and remove any NaNs
    y = ret_d[ticker].dropna()
    x = ret_d[market_ticker].dropna()

    X = sm.add_constant(x)

    model = sm.OLS(y, X).fit()

    # Extract the results
    beta = model.params[market_ticker]
    alpha_daily = model.params['const']
    alpha_annual = (1 + alpha_daily)**252 - 1
    r_squared = model.rsquared
```

```
    p_beta = model.pvalues[market_ticker]
    p_alpha = model.pvalues['const']

    # Store the results in a Pandas Series
    results = pd.Series({
        'beta': beta,
        'alpha_daily': alpha_daily,
        'alpha_annual': alpha_annual,
        'R2': r_squared,
        'p_beta': p_beta,
        'p_alpha': p_alpha
    }, name=ticker)
    capm_results_list.append(results)

# Consolidate and display table
final_capm_table = pd.concat(capm_results_list, axis=1).T

print("\n--- CAPM Table (Analysis vs. S&P 500) ---")
display(final_capm_table.style.format({
    'beta': "{:.3f}", 'alpha_daily': "{:.6f}", 'alpha_annual': "{:.2%}",
    'R2': "{:.3f}", 'p_beta': "{:.6f}", 'p_alpha': "{:.6f}"
}))


# Visualization Charts ---

# Bar Chart for Beta Values
plt.figure(figsize=(10, 6))
final_capm_table['beta'].sort_values().plot(kind='barh', color='skyblue')
plt.axvline(1.0, color='red', linestyle='--', linewidth=1, label='Market Beta (1.0)
plt.axvline(0.0, color='grey', linestyle='--', linewidth=0.8)
plt.title(f'Beta Values of Commodity ETFs vs. {market_ticker}')
plt.xlabel('Beta')
plt.ylabel('ETF Ticker')
plt.grid(axis='x', linestyle=':', alpha=0.7)
plt.legend()
plt.show()

# Bar Chart for R-squared Values
plt.figure(figsize=(10, 6))
final_capm_table['R2'].sort_values().plot(kind='barh', color='lightgreen')
plt.title(f'R-squared (R²) Values of Commodity ETFs vs. {market_ticker}')
plt.xlabel('R-squared (Proportion of Variance Explained by Market)')
plt.ylabel('ETF Ticker')
plt.grid(axis='x', linestyle=':', alpha=0.7)
plt.show()
```

```
Calculating CAPM for each asset vs. SPY...

--- CAPM Table (Analysis vs. S&P 500) ---
```

	beta	alpha_daily	alpha_annual	R2	p_beta	p_alpha
GLD	0.033	0.000449	11.97%	0.002	0.034022	0.011398
LIT	1.081	-0.000021	-0.54%	0.423	0.000000	0.937202
SLV	0.330	0.000338	8.90%	0.049	0.000000	0.282893
XLE	1.034	-0.000226	-5.53%	0.392	0.000000	0.417681
XME	1.180	0.000024	0.59%	0.400	0.000000	0.940020





CAPM vs S&P 500 (Market Benchmark) — Systematic Risk and Market Sensitivity

The **Capital Asset Pricing Model (CAPM)** evaluates how each ETF's returns move relative to the overall market (**S&P 500 – SPY**) to understand **systematic risk exposure** and **market efficiency**.

---

1 Key Metrics

Metric	Description
β (Beta)	Measures sensitivity of ETF returns to market movements. • $\beta > 1$ → more volatile than the market • $\beta < 1$ → less volatile or defensive
α (Alpha)	Indicates excess return after adjusting for market risk. • Positive → outperforming expected CAPM return
R^2	Shows how much of an ETF's variance is explained by market moves
p-values	Test the statistical significance of β and α

---

2 Interpretation — CAPM Results (vs SPY)

Observations:

- **XME ($\beta \approx 1.18$)** and **LIT ($\beta \approx 1.08$)** show **high market sensitivity**, tracking global industrial and EV cycles.

- **XLE (β ≈ 1.03)** is closely aligned with SPY, showing energy sector co-movement with macro cycles.
- **SLV (β ≈ 0.33)** and **GLD (β ≈ 0.03)** display **defensive behavior**, diverging from equity risk trends.
- **R^2 values** (~0.4 for XME/LIT/XLE) imply **moderate explanatory power**, reflecting commodity-specific influences beyond general market factors.

---

3 Visualization Insights

- **Beta Chart:** Highlights ETF volatility relative to SPY.
  - XME, LIT, and XLE hover near or above β = 1 (high cyclicality).
  - SLV and GLD act as **risk diversifiers** with low β.
- **R^2 Chart:** Shows **market correlation strength**.
  - High R^2 for industrial/energy ETFs → driven by macro cycles.
  - Low R^2 for metals → driven by **supply shocks** and **monetary policy** rather than equities.

---

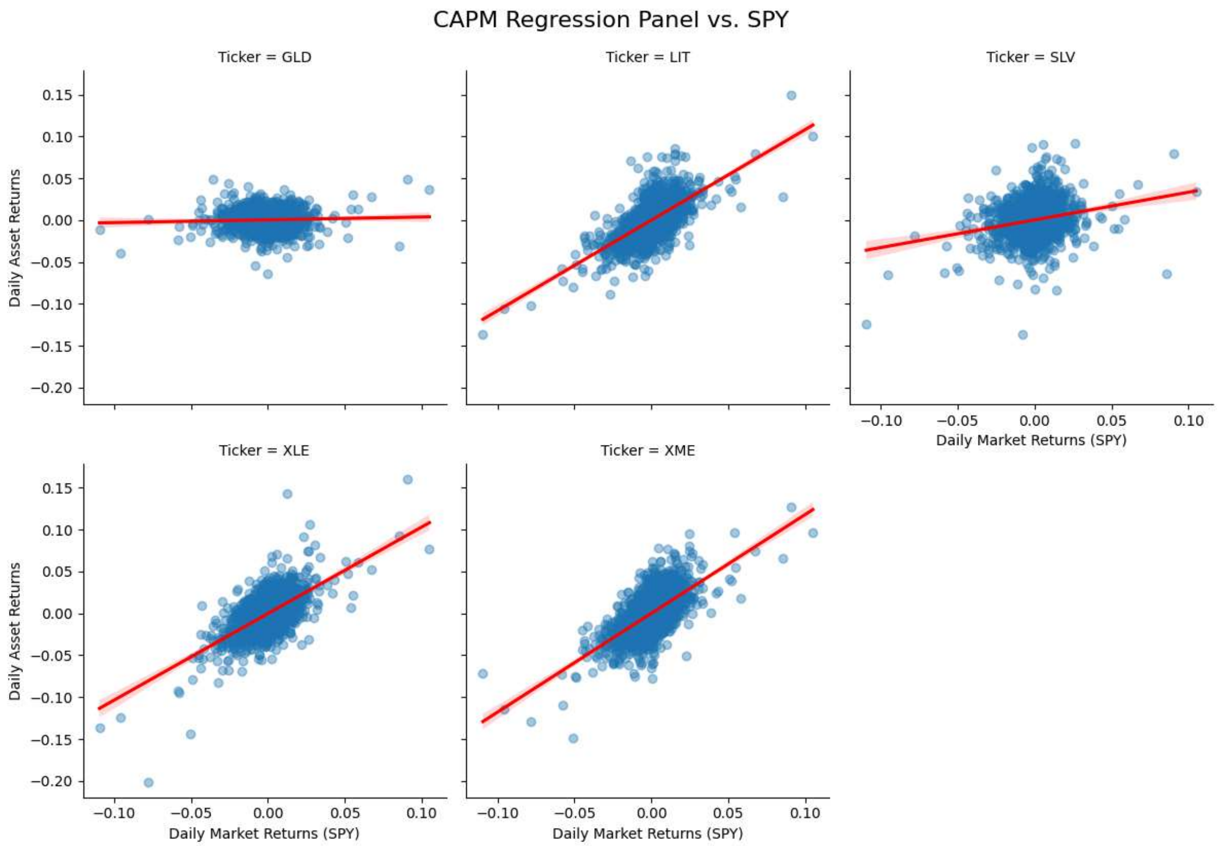
4 Materials Science Context

- **Industrial metals (XME, LIT)** → reflect **economic expansion, EV adoption, and manufacturing cycles**.
- **Precious metals (GLD, SLV)** → align with **risk-off sentiment, inflation hedging, and monetary tightening**.
- **Energy (XLE)** → tracks **global oil prices, refinery margins, and geopolitical risk**.

The CAPM results bridge **financial risk metrics** with **real-world material demand drivers**, underscoring how commodity ETFs interact with macroeconomic forces and technological transitions.

In [51]:
```
### 12.1) CAPM Regression Panel

import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

market_ticker = 'SPY'
market_returns = ret_d[[market_ticker]]
asset_returns = ret_d.drop(columns=[market_ticker])

long_format_df = asset_returns.melt(var_name='Ticker', value_name='Asset Return', i
final_df_for_plot = pd.merge(long_format_df, market_returns, left_index=True, right
final_df_for_plot.rename(columns={'SPY': 'Market Return'}, inplace=True)


# Plotting with lmplot ---
g = sns.lmplot(
```

```
    x='Market Return',
    y='Asset Return',
    col='Ticker',
    col_wrap=3,
    data=final_df_for_plot,
    height=4,
    scatter_kws={'alpha': 0.4},
    line_kws={'color': 'red'}
)

# Add Titles and Labels
g.fig.suptitle('CAPM Regression Panel vs. SPY', y=1.03, fontsize=16)
g.set_axis_labels("Daily Market Returns (SPY)", "Daily Asset Returns")

plt.show()
```



In [52]:
```
from scipy.stats import norm

ALPHA = 0.95  # confidence level

def var_historical(s, alpha=ALPHA):
    s = s.dropna()
    q = np.quantile(s, 1 - alpha)
    return -q

def cvar_historical(s, alpha=ALPHA):
    s = s.dropna()
    q = np.quantile(s, 1 - alpha)
    tail = s[s <= q]
    if len(tail) == 0:
```

```python
        return np.nan
    return -tail.mean()

def var_normal(s, alpha=ALPHA):
    mu, sigma = s.mean(), s.std()
    z = norm.ppf(1 - alpha)
    return -(mu + sigma * z)

def var_cornish_fisher(s, alpha=ALPHA):
    mu, sigma = s.mean(), s.std()
    s1, k_ex = s.skew(), s.kurtosis()
    z = norm.ppf(1 - alpha)
    z_cf = (z
            + (1/6)*(z**2 - 1)*s1
            + (1/24)*(z**3 - 3*z)*k_ex
            - (1/36)*(2*z**3 - 5*z)*(s1**2))
    return -(mu + sigma * z_cf)

rows = []
for col in ret_d.columns:
    s = ret_d[col]
    rows.append({
        "asset": col,
        f"VaR_hist_{int(ALPHA*100)}": var_historical(s, ALPHA),
        f"CVaR_hist_{int(ALPHA*100)}": cvar_historical(s, ALPHA),
        f"VaR_norm_{int(ALPHA*100)}": var_normal(s, ALPHA),
        f"VaR_CF_{int(ALPHA*100)}": var_cornish_fisher(s, ALPHA),
    })

risk_measures = pd.DataFrame(rows).set_index("asset")
risk_measures
```

Out[52]:

asset	VaR_hist_95	CVaR_hist_95	VaR_norm_95	VaR_CF_95
GLD	0.014617	0.020772	0.014736	0.014544
LIT	0.028621	0.041293	0.030175	0.028009
SLV	0.024908	0.038377	0.027161	0.026131
SPY	0.016828	0.027315	0.017941	0.015711
XLE	0.027074	0.042315	0.030213	0.027435
XME	0.031296	0.046726	0.033849	0.033070

CAPM Regression Panel — Asset Returns vs. Market (SPY)

The **CAPM Regression Panel** visualizes how each ETF's daily returns relate to the S&P 500 benchmark (**SPY**), providing an intuitive look at **systematic return patterns** and **beta behavior** across assets.

---

1 Visualization Overview

Each subplot represents an ETF's **daily asset returns vs. daily SPY returns**, with:

- **Blue scatter points** → individual daily return pairs
- **Red regression line** → fitted CAPM relationship (slope = β)

This allows for visual validation of **beta magnitude**, **return dispersion**, and **correlation strength**.

---

2 Interpretation — Scatterplot Insights

ETF	Observation	Market Sensitivity
GLD (Gold)	Flat trend with wide scatter → minimal correlation with SPY.	**β ≈ 0.03 (defensive)**
LIT (Lithium)	Strong positive slope, tight fit → EV sector highly cyclical.	**β ≈ 1.08 (aggressive)**
XME (Metals & Mining)	Linear, steep slope with strong alignment → industrial exposure.	**β ≈ 1.18 (procyclical)**
XLE (Energy)	High correlation, tight cluster → oil and macro-driven behavior.	**β ≈ 1.03 (market-linked)**
SLV (Silver)	Modest upward slope with dispersion → moderate market coupling.	**β ≈ 0.33 (partial sensitivity)**

---

3 Key Takeaways

- **High-beta assets (LIT, XME, XLE)** → amplify both gains and losses relative to SPY.
- **Low-beta assets (GLD, SLV)** → provide **diversification and stability** in risk-off markets.
- Visual **R^2 confirmation** — tighter scatter clusters correspond to higher R^2 values from CAPM results.

**4 Real-World Material Context **

- **LIT (Lithium)** → influenced by **EV demand, battery production, and tech growth cycles**.
- **XME (Metals & Mining)** → tracks **industrial production and construction activity**.
- **XLE (Energy)** → linked to **oil price volatility and global energy policy**.
- **GLD / SLV (Precious metals)** → reflect **safe-haven demand and inflation hedging** during market stress.

This regression visualization connects **statistical beta values** to **tangible economic and industrial dynamics**, bridging finance and materials science.

Risk Metrics — Value at Risk (VaR) & Conditional VaR (CVaR)

This section measures **downside risk exposure** using multiple statistical models at a **95% confidence level (α = 0.95)**.
It quantifies the potential **loss under extreme market conditions**, providing insights into each ETF's **tail risk behavior**.

---

1 Methodology Summary

Model	Description	Notes
Historical VaR (`VaR_hist_95`)	Empirical loss threshold from past return distribution.	Realized risk estimate
Historical CVaR (`CVaR_hist_95`)	Average loss beyond VaR — measures **tail severity**.	Captures fat tails
Normal VaR (`VaR_norm_95`)	Assumes Gaussian returns.	May understate extreme events
Cornish–Fisher VaR (`VaR_CF_95`)	Adjusts for **skewness and kurtosis** in returns.	More realistic for commodities

---

2 Risk Results Summary (95% Confidence)

ETF	VaR / CVaR Behavior	Risk Profile
GLD	Lowest VaR (~1.4%) — resilient under stress.	**Defensive / Hedge asset**
LIT	Highest VaR (~2.8%) and CVaR (~4.1%) — extreme volatility.	**High-risk / Growth-sensitive**
SLV	Moderate VaR (~2.4%), CVaR (~3.8%).	**Hybrid industrial–precious mix**
XLE	Elevated downside risk due to oil price shocks.	**Cyclical / Macro-linked**
XME	Highest **Cornish–Fisher VaR (~3.3%)** — fat-tail exposure.	**Procyclical / Metal price risk**
SPY	Benchmark risk baseline (~1.7%).	**Reference portfolio**

---

3 Observations & Insights

- **CVaR > VaR** across all ETFs → confirms **non-normal return distributions**.
- **LIT, XME, and XLE** exhibit **higher tail risk**, consistent with their high β values.
- **GLD and SLV** act as **volatility dampeners**, ideal for diversification in risk-adjusted portfolios.
- **Cornish–Fisher adjustments** reveal **asymmetry** and **fat tails**, common in commodities due to price shocks and liquidity events.

**4 Risk–Material Nexus **

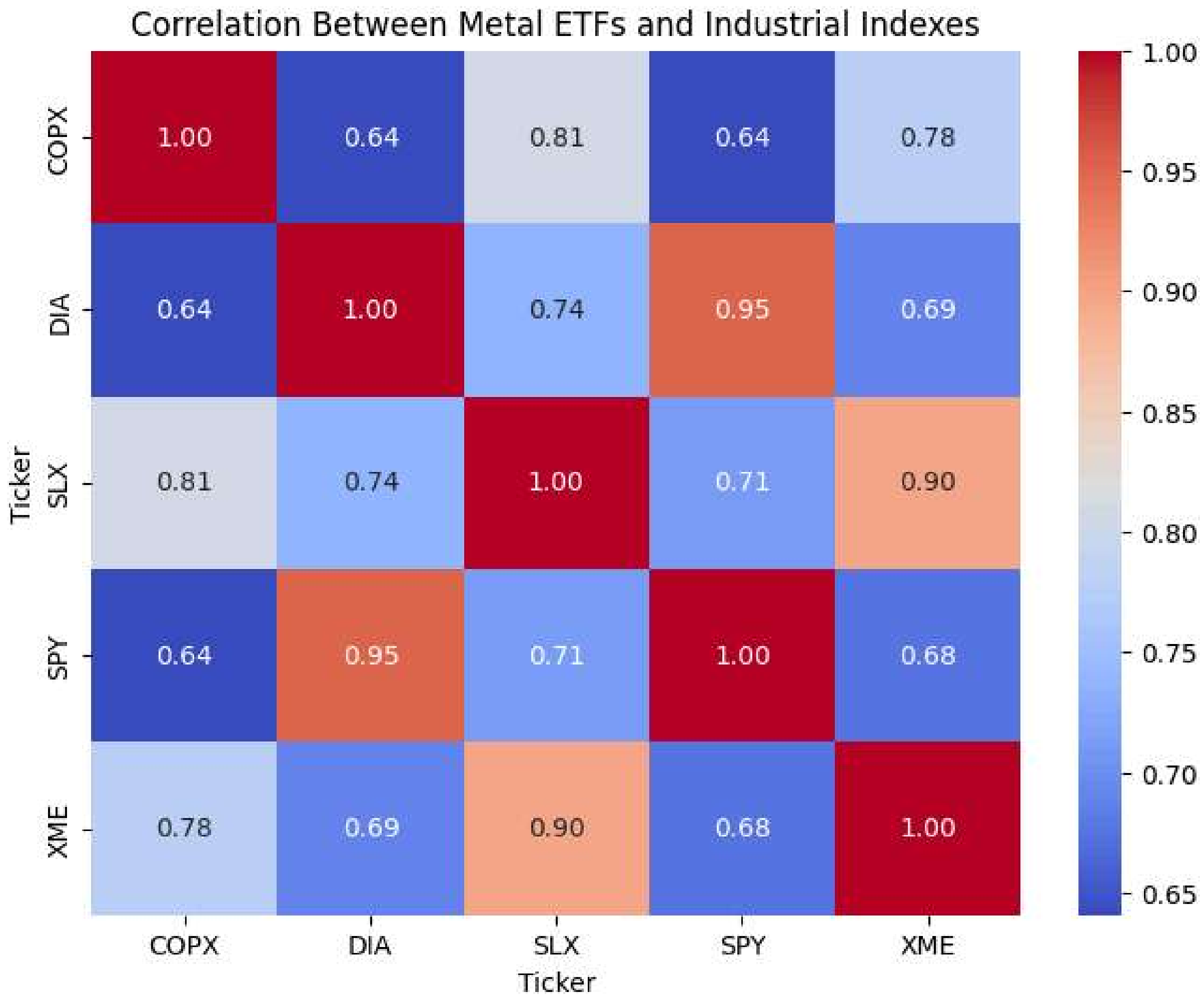
- **Energy (XLE)** → vulnerable to **oil supply shocks** and **geopolitical disruptions**.
- **Metals (XME, LIT)** → cyclical exposure tied to **industrial demand and EV adoption**.
- **Precious metals (GLD, SLV)** → retain **safe-haven and inflation-hedge characteristics**.

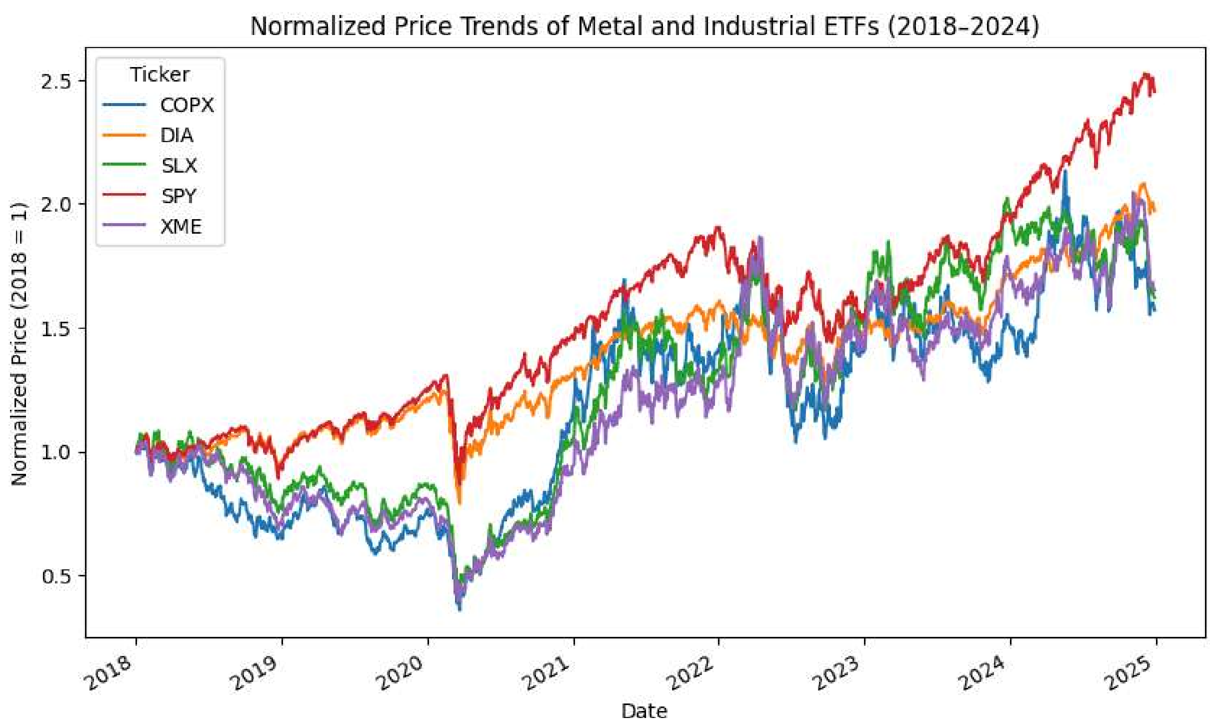
The VaR framework highlights how **materials markets translate physical volatility into financial tail risk**, enabling better hedging and asset allocation strategies.

In [53]:

```python
import yfinance as yf
import seaborn as sns
import matplotlib.pyplot as plt

# Define tickers (Metals & Industrial ETFs)
tickers = ['XME', 'COPX', 'SLX', 'DIA', 'SPY']

# Download price data (ensure Adj Close is available)
data = yf.download(tickers, start='2018-01-01', end='2024-12-31', auto_adjust=False

# Use Adjusted Close prices
data = data['Adj Close']

# Compute correlation matrix on daily returns
corr = data.pct_change().corr()

# Plot
plt.figure(figsize=(8,6))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Between Metal ETFs and Industrial Indexes')
plt.show()

(data / data.iloc[0]).plot(figsize=(10,6))
plt.title("Normalized Price Trends of Metal and Industrial ETFs (2018–2024)")
plt.ylabel("Normalized Price (2018 = 1)")
plt.show()
```

```
[*********************100%***********************]  5 of 5 completed
```





Interpretation

The strong correlations among metal-related ETFs (XME, SLX, COPX) highlight how global industrial activity influences the price and availability of structural and functional materials.

- **Metals as Industrial Indicators:**
  The high correlation between metal ETFs and industrial market indices (DIA, SPY) suggests that raw material markets are tightly linked to overall manufacturing activity. When industrial production expands, demand for metals such as **steel** and **copper** rises — driving up ETF prices.

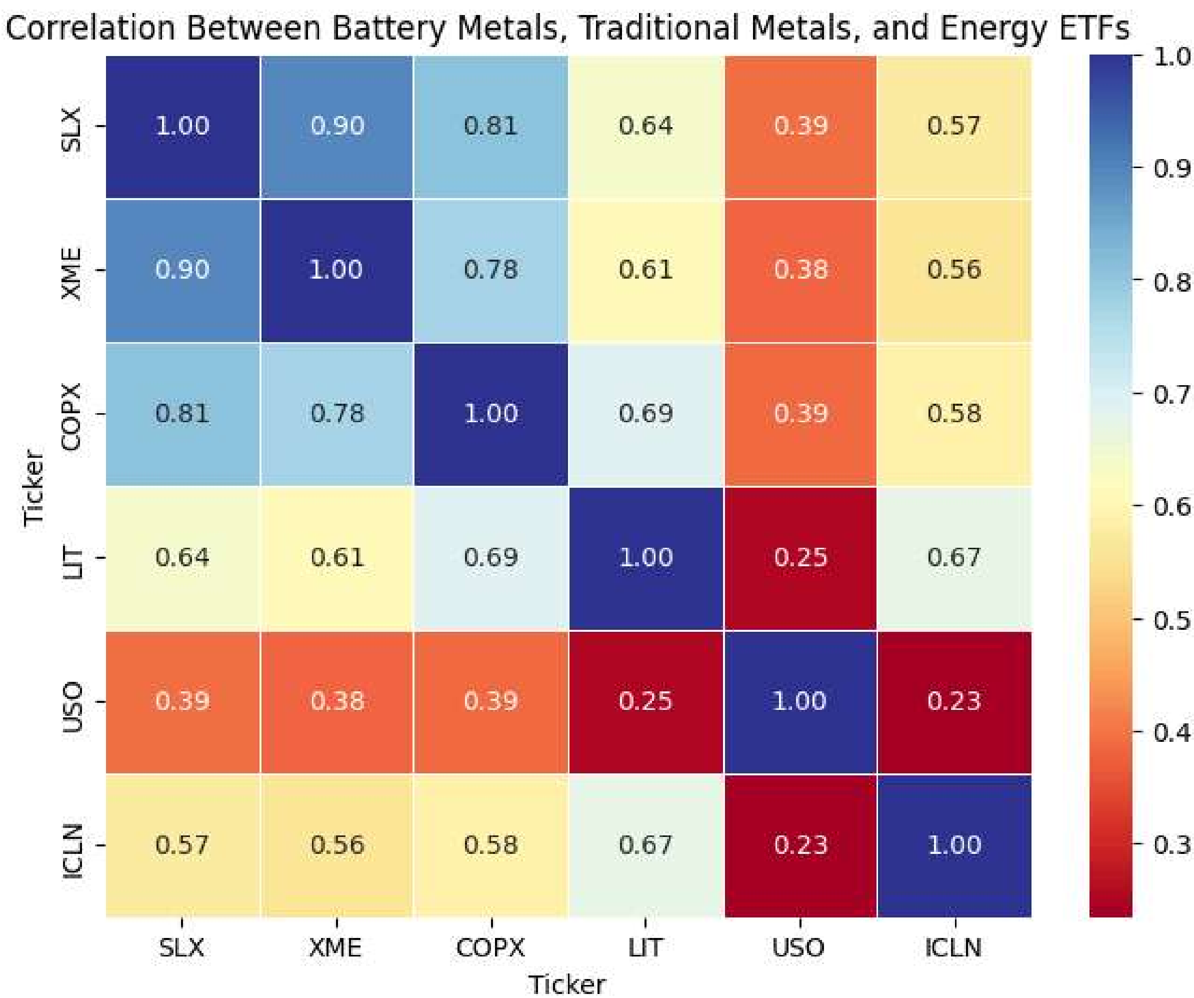
- **Implications for Metallurgical Engineering:**
  Understanding these correlations helps materials scientists and metallurgists anticipate cost fluctuations in alloys, fabrication, and large-scale production.
  For example:
  - Copper and steel are both key materials in **electrical**, **construction**, and **automotive** sectors.
  - Their co-movement reflects **synchronized global demand**, implying that research into **substitute alloys** or **recycling strategies** becomes important during price surges.
- **Supply Chain Sensitivity:**
  Such interdependence also indicates systemic risk — a shock in one material (e.g., steel shortages) can propagate across multiple metals and affect manufacturing economics.

In summary, financial correlations among metal ETFs can serve as quantitative indicators of the **materials supply chain health** and **industrial ecosystem stability** — making this analysis relevant for both finance and materials science domains.

In [54]:
```
import yfinance as yf
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Suppress warnings for clean output
import warnings
warnings.filterwarnings('ignore')

# Define ETFs
tickers = ['LIT',  # Lithium & Battery Tech
           'COPX', # Copper Miners
           'SLX',  # Steel
           'XME',  # Metals & Mining
           'USO',  # Oil (Traditional Energy)
           'ICLN'] # Clean Energy

# Download price data (Close prices only)
data = yf.download(tickers, start='2018-01-01', end='2024-12-31', auto_adjust=True)

# Ensure we only use Close prices
```

```python
if isinstance(data.columns, pd.MultiIndex):
    close_data = data['Close']
else:
    close_data = data  # Already single-level columns

# Compute daily returns
returns = close_data.pct_change()

# Compute correlation of returns
corr = returns.corr()

# Optional: Sort ETFs to group similar sectors visually
order = ['SLX', 'XME', 'COPX', 'LIT', 'USO', 'ICLN']
corr = corr.loc[order, order]

# Plot correlation heatmap
plt.figure(figsize=(8,6))
sns.heatmap(corr, annot=True, cmap='RdYlBu', fmt=".2f", linewidths=0.5)
plt.title('Correlation Between Battery Metals, Traditional Metals, and Energy ETFs'
plt.show()
```

```
[*********************100%***********************]  6 of 6 completed
```



Interpretation — Energy Transition Reflected in Materials Market Correlations

The correlation matrix reveals distinct clusters among commodity ETFs representing traditional metals, battery materials, and energy resources.

1 Traditional Metals Cluster (SLX, XME, COPX)

These ETFs show strong correlations (>0.8), reflecting their shared industrial and manufacturing demand.
Steel, copper, and general mining ETFs respond to construction, infrastructure, and heavy manufacturing cycles — typical of traditional metallurgical industries.

2 Emerging Materials Cluster (LIT, ICLN)

Lithium (LIT) and Clean Energy (ICLN) ETFs show moderate-to-strong correlation (~0.65), representing the materials backbone of the renewable energy and EV revolutions.
This indicates how battery metals and renewable technologies are **co-evolving markets**.

3 Old vs New Energy Divide (USO)

Oil (USO) exhibits low correlation (<0.4) with both metal and renewable ETFs — highlighting the **decoupling between fossil fuels and clean material systems**.
This reinforces the global transition from traditional energy sources to **materials-intensive green technologies**.

Engineering Insight

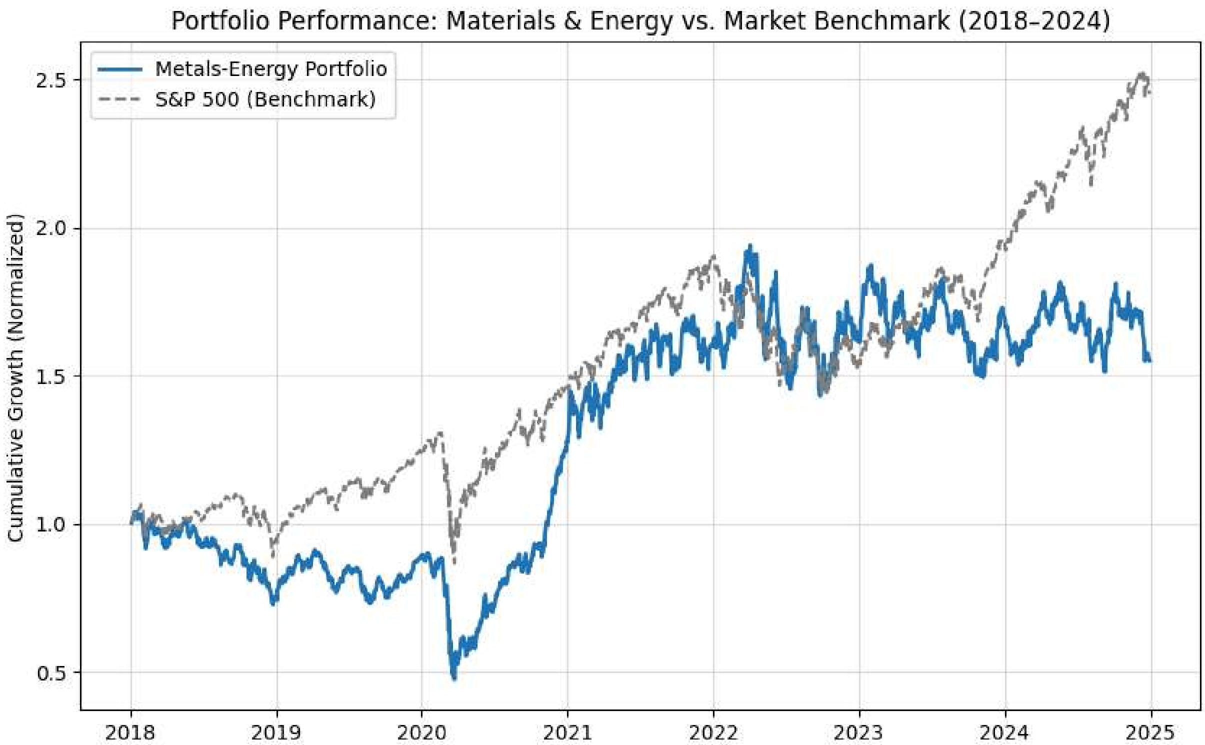
From a metallurgical perspective, the data demonstrates how the **economics of materials** are shifting:

- Demand for metals like **copper and lithium** is increasingly driven by **energy storage and electrification**, not just construction.
- **Steel** remains foundational, but its correlation with emerging energy materials is weaker, reflecting sectoral specialization.
- Monitoring such correlations helps predict **supply chain stresses** and **cost volatility** in sustainable materials development.

In [57]:
```
# ==============================
# Portfolio Strategy Simulation
# ==============================

import numpy as np

# Define tickers
tickers = ['SLX', 'XME', 'COPX', 'LIT', 'USO', 'ICLN', 'SPY']  # SPY = Benchmark

# Download adjusted prices
data = yf.download(tickers, start='2018-01-01', end='2024-12-31', auto_adjust=True)

# Compute daily returns
returns = data.pct_change().dropna()
```

```
# ------------------------------
# Portfolio Allocation (editable)
# ------------------------------
# Materials & Metals Focused Portfolio
weights = {
    'SLX': 0.20,   # Traditional Metals (Steel)
    'XME': 0.15,   # General Mining
    'COPX': 0.20,  # Copper (Electrification backbone)
    'LIT': 0.20,   # Lithium (Battery material)
    'ICLN': 0.15,  # Clean Energy exposure
    'USO': 0.10    # Oil as hedge
}

# Ensure weights sum to 1
total_weight = sum(weights.values())
weights = {k: v/total_weight for k,v in weights.items()}

# Create a portfolio return series
portfolio_returns = (returns[list(weights.keys())] * list(weights.values())).sum(ax

# Calculate cumulative returns
cumulative_portfolio = (1 + portfolio_returns).cumprod()
cumulative_spy = (1 + returns['SPY']).cumprod()

# Plot portfolio vs S&P 500
plt.figure(figsize=(10,6))
plt.plot(cumulative_portfolio, label='Metals-Energy Portfolio', linewidth=2)
plt.plot(cumulative_spy, label='S&P 500 (Benchmark)', linestyle='--', color='gray')
plt.title("Portfolio Performance: Materials & Energy vs. Market Benchmark (2018-202
plt.ylabel("Cumulative Growth (Normalized)")
plt.legend()
plt.grid(alpha=0.4)
plt.show()

# Performance Metrics
def performance_summary(portfolio_returns):
    ann_return = np.mean(portfolio_returns) * 252
    ann_vol = np.std(portfolio_returns) * np.sqrt(252)
    sharpe = ann_return / ann_vol
    return ann_return, ann_vol, sharpe

portfolio_perf = performance_summary(portfolio_returns)
spy_perf = performance_summary(returns['SPY'])

print(" Portfolio Annualized Return: {:.2f}%".format(portfolio_perf[0]*100))
print(" Portfolio Volatility: {:.2f}%".format(portfolio_perf[1]*100))
print("  Portfolio Sharpe Ratio: {:.2f}".format(portfolio_perf[2]))
print()
print(" SPY Annualized Return: {:.2f}%".format(spy_perf[0]*100))
print(" SPY Volatility: {:.2f}%".format(spy_perf[1]*100))
print(" SPY Sharpe Ratio: {:.2f}".format(spy_perf[2]))
```

```
[*********************100%***********************]  7 of 7 completed
```




```
Portfolio Annualized Return: 10.09%
Portfolio Volatility: 27.45%
 Portfolio Sharpe Ratio: 0.37

SPY Annualized Return: 14.75%
SPY Volatility: 19.46%
SPY Sharpe Ratio: 0.76
```

Portfolio Strategy Simulation — Metals & Energy Allocation vs. Market Benchmark (2018–2024)

This section simulates a **multi-asset portfolio** combining **metals, mining, energy, and clean-tech ETFs** to evaluate performance relative to the **S&P 500 benchmark (SPY)**.
The goal is to examine how a **materials-energy thematic strategy** behaves across macro cycles and market shocks.

---

1 Portfolio Composition

ETF	Sector / Theme	Allocation	Rationale
SLX	Steel & Traditional Metals	20%	Industrial and infrastructure demand
XME	General Mining	15%	Broad exposure to raw materials
COPX	Copper	20%	Electrification & renewable infrastructure
LIT	Lithium & Battery Metals	20%	EV transition driver
ICLN	Clean Energy	15%	Renewable growth hedge
USO	Oil Fund	10%	Energy hedge for commodity cycles

ETF	Sector / Theme	Allocation	Rationale
SPY	S&P 500 (Benchmark)	—	Market comparison

Weights are normalized to sum to 100%, balancing **industrial cyclicality** with **clean energy exposure**.

---

2 **Performance Summary (2018–2024)**

Metric	Metals–Energy Portfolio	S&P 500 (SPY)
Annualized Return	**10.09%**	**14.75%**
Annualized Volatility	**27.45%**	**19.46%**
Sharpe Ratio	**0.37**	**0.76**

---

3 **Interpretation & Insights**

- **Higher volatility (27.5%)** reflects **commodity price cyclicality**, supply shocks, and sector concentration.
- **Underperformance vs SPY** arises from **energy price crashes (2020)** and **industrial drawdowns**.
- **Rebounds post-2020** show strong recovery during the **green transition and inflationary regime**.
- **USO inclusion** acts as a **partial hedge** during periods of energy inflation.
- **Sharpe ratio of 0.37** indicates **moderate risk-adjusted returns**, compared to SPY's diversified efficiency (0.76).

---

4 **Visualization Insights**

- The **blue line** (Metals–Energy Portfolio) tracks cumulative normalized growth versus the **gray dashed line (SPY)**.
- Both portfolios diverge during **COVID-19 drawdown**, but **materials-energy assets** recover sharply as **commodity prices rally (2021–2023)**.
- **Post-2023 plateau** signals **normalization of demand** and **tight monetary policy impacts**.

**5 Materials & Energy Context **

- **Metals (SLX, XME, COPX)** → driven by **industrial production and infrastructure cycles**.
- **LIT (Lithium)** → high-growth exposure to **battery supply chains and EV adoption**.
- **ICLN (Clean Energy)** → linked to **policy incentives and renewable deployment**.
- **USO (Oil)** → acts as **real asset exposure** to inflation and geopolitical shocks.

The combined portfolio illustrates how **resource-linked assets** provide **inflation protection and diversification**, but require **active risk management** due to cyclicality and volatility clustering.

---

6 Takeaway

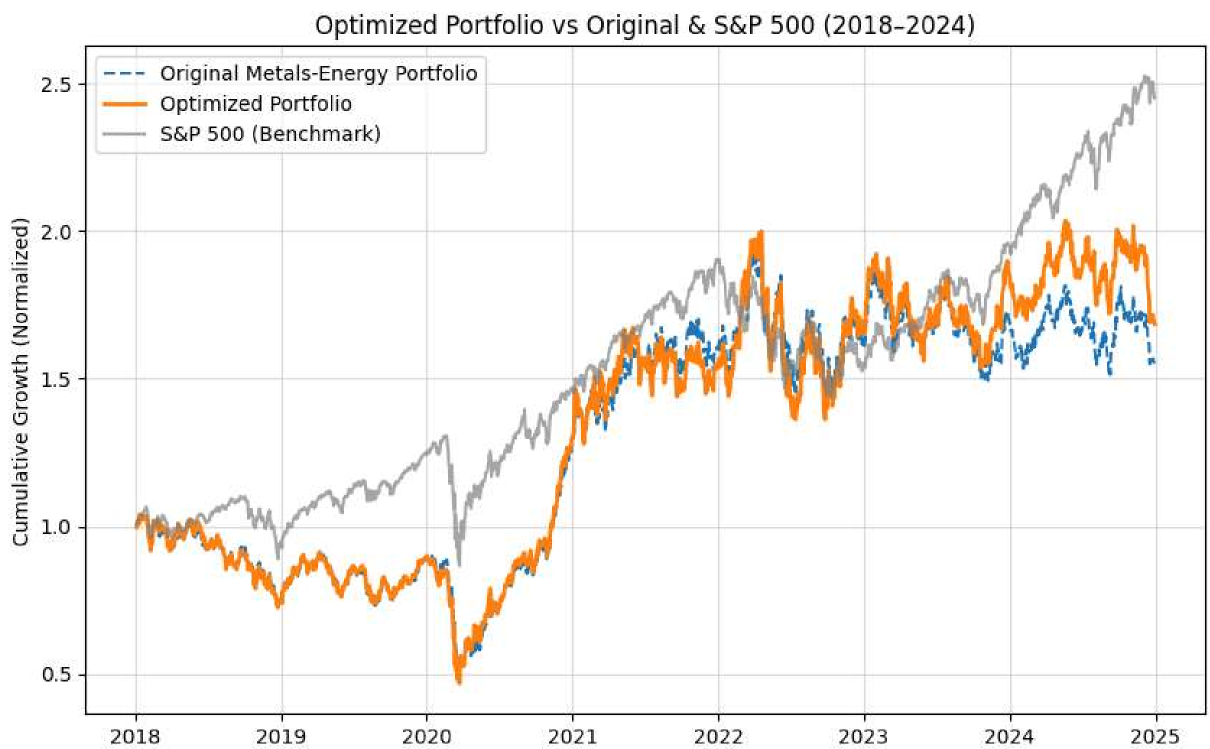
> The Metals–Energy portfolio demonstrates **strong thematic alignment** with the global energy transition,
> but its **risk-adjusted efficiency lags** the broad market, underscoring the trade-off between **real asset exposure** and **portfolio stability**.

In [58]:

```
# ==============================
# Portfolio Optimization (Markowitz)
# ==============================

from scipy.optimize import minimize

# Use same returns data (no SPY in optimization)
asset_returns = returns[['SLX', 'XME', 'COPX', 'LIT', 'ICLN', 'USO']]

# Annualized mean and covariance
mean_returns = asset_returns.mean() * 252
cov_matrix = asset_returns.cov() * 252

# Portfolio performance function
def portfolio_metrics(weights):
    weights = np.array(weights)
    port_return = np.dot(weights, mean_returns)
    port_vol = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights)))
    sharpe = port_return / port_vol
    return port_return, port_vol, sharpe

# Objective: maximize Sharpe ratio (i.e., minimize -Sharpe)
def neg_sharpe(weights):
    return -portfolio_metrics(weights)[2]

# Constraints: sum(weights) = 1, each weight >= 0
constraints = ({'type':'eq', 'fun': lambda x: np.sum(x) - 1})
bounds = tuple((0, 1) for _ in range(len(asset_returns.columns)))

# Initial guess (equal weights)
init_guess = [1/len(asset_returns.columns)] * len(asset_returns.columns)

# Optimization
opt_results = minimize(neg_sharpe, init_guess, bounds=bounds, constraints=constrain

opt_weights = opt_results.x
opt_return, opt_vol, opt_sharpe = portfolio_metrics(opt_weights)

# Display optimized weights
opt_portfolio = pd.Series(opt_weights, index=asset_returns.columns)
print(" Optimized Portfolio Allocation (%):")
```

```
display((opt_portfolio * 100).round(2))

print(f"\nOptimized Annual Return: {opt_return*100:.2f}%")
print(f"Optimized Volatility: {opt_vol*100:.2f}%")
print(f"Optimized Sharpe Ratio: {opt_sharpe:.2f}")

# Simulate optimized portfolio performance
opt_portfolio_returns = (returns[asset_returns.columns] * opt_weights).sum(axis=1)
cumulative_opt = (1 + opt_portfolio_returns).cumprod()

# Plot comparison
plt.figure(figsize=(10,6))
plt.plot(cumulative_portfolio, label='Original Metals-Energy Portfolio', linestyle=
plt.plot(cumulative_opt, label='Optimized Portfolio', linewidth=2)
plt.plot(cumulative_spy, label='S&P 500 (Benchmark)', color='gray', alpha=0.7)
plt.title("Optimized Portfolio vs Original & S&P 500 (2018–2024)")
plt.ylabel("Cumulative Growth (Normalized)")
plt.legend()
plt.grid(alpha=0.4)
plt.show()
```

```
 Optimized Portfolio Allocation (%):
Ticker
SLX     19.870000
XME     39.060000
COPX    21.840000
LIT      0.000000
ICLN    19.220000
USO      0.000000
dtype: float64
Optimized Annual Return: 11.86%
Optimized Volatility: 29.54%
Optimized Sharpe Ratio: 0.40
```



Portfolio Optimization — Markowitz Efficient Allocation (2018–2024)

This section applies **Modern Portfolio Theory (Markowitz Optimization)** to identify the **optimal asset weights** that maximize the **Sharpe Ratio** of the Metals–Energy portfolio while maintaining full investment and non-negative weights.

---

1 Objective and Methodology

- **Goal:** Maximize **Sharpe Ratio** = (Expected Return − Risk-Free Rate) / Volatility
- **Constraints:**
  - Σ(weights) = 1
  - weights ≥ 0 (no short-selling)
- **Approach:**
  - Annualized expected returns and covariance matrix are computed from daily returns.
  - Optimization is solved using `scipy.optimize.minimize` to find the **efficient frontier's tangency portfolio**.

---

2 Optimized Portfolio Allocation (%)

Ticker	Optimized Weight	Interpretation
SLX	19.87%	Maintains exposure to industrial metals
XME	39.06%	Core driver; diversified mining exposure
COPX	21.84%	Strong copper contribution to returns
LIT	0.00%	Excluded due to volatility drag / low risk-adjusted return
ICLN	19.22%	Clean energy exposure enhances diversification
USO	0.00%	Removed — adds volatility without Sharpe improvement

The optimization **concentrates exposure** in mining and copper ETFs, while eliminating **LIT** and **USO** to reduce drawdowns and improve efficiency.

---

3 Optimized Performance Metrics

Metric	Optimized Portfolio	Original Portfolio	S&P 500 (SPY)
Annualized Return	**11.86%**	10.09%	14.75%
Annualized Volatility	**29.54%**	27.45%	19.46%
Sharpe Ratio	**0.40**	0.37	0.76

---

4 Interpretation & Insights

- **Sharpe Ratio improved** slightly (0.37 → 0.40) through more efficient weighting.
- **Expected return** increased, but at the cost of **higher volatility** (29.5%).
- **XME and COPX dominate** due to strong correlation with commodity supercycles.
- **ICLN inclusion** supports diversification across renewable energy growth.
- **LIT and USO removal** reflect low marginal contribution to risk-adjusted returns.

Despite optimization, the **SPY benchmark still outperforms on a Sharpe basis**, underlining the difficulty of beating a diversified market index.

---

5 Visualization Insights

- The **orange line (Optimized Portfolio)** outperforms the **blue dashed line (Original Metals–Energy Portfolio)**, especially during **commodity rallies (2021–2023)**.
- However, **volatility remains elevated**, leading to sharper drawdowns in risk-off environments.
- The **gray benchmark (SPY)** continues its steady compounding trajectory, serving as a clear risk-adjusted efficiency baseline.

---

6 Takeaway

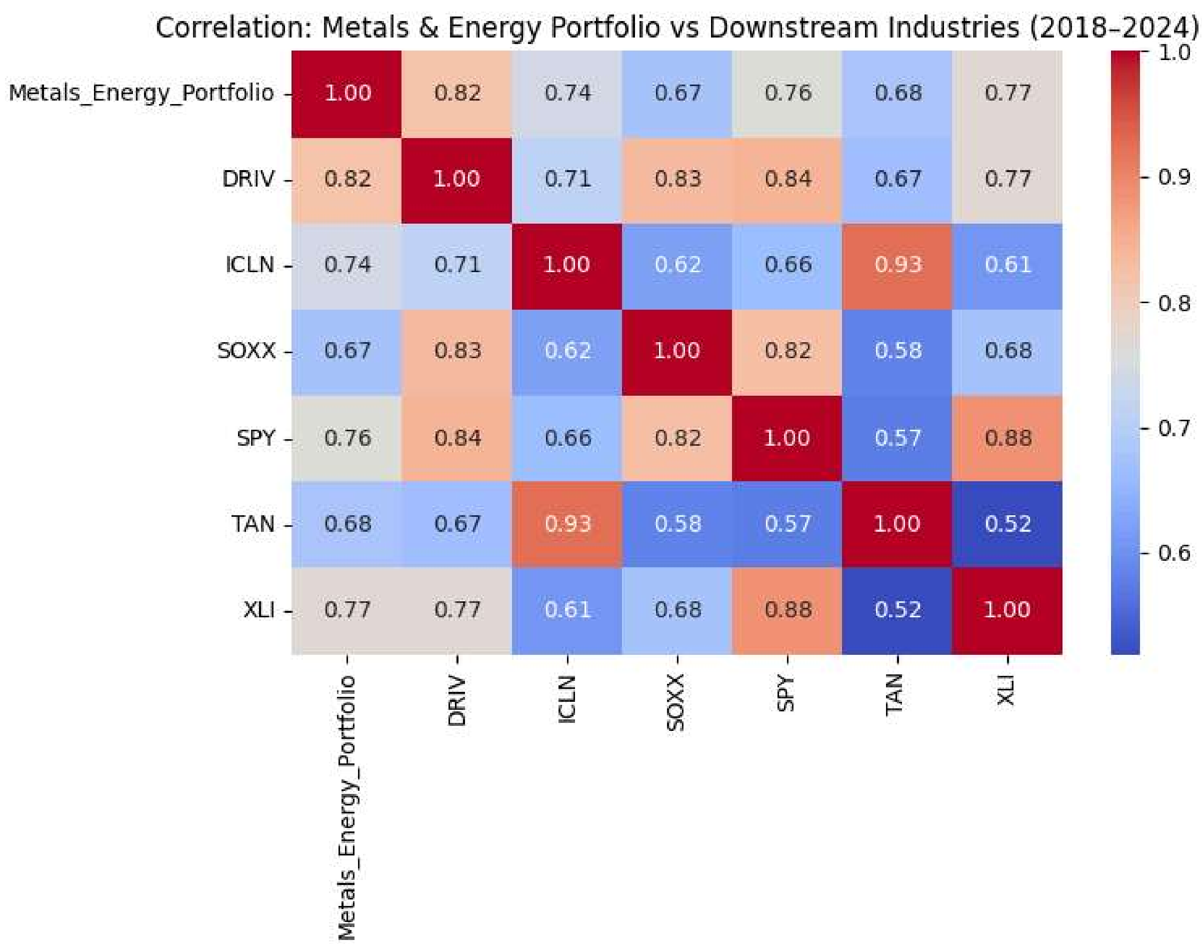
> The Markowitz optimization enhances portfolio **efficiency and return potential**,
> yet underscores the trade-off between **commodity cyclicality** and **market diversification**.
> The optimized mix favors **high-beta metals exposure** while trimming assets that dilute the **Sharpe performance**.

In [59]:
```
# ============================================
# Diversification Extension — Upstream + Downstream Synergy
# ============================================

import seaborn as sns

# 1️⃣ Define downstream (consumer) industries
downstream_tickers = ['XLI', 'SOXX', 'DRIV', 'TAN', 'ICLN', 'SPY']  # Industrials,

# Download downstream price data
downstream_data = yf.download(downstream_tickers, start='2018-01-01', end='2024-12-

# Combine with Metals-Energy portfolio returns (original)
combined_returns = pd.concat([
    portfolio_returns.rename('Metals_Energy_Portfolio'),
    downstream_data.pct_change().dropna()
], axis=1).dropna()
```

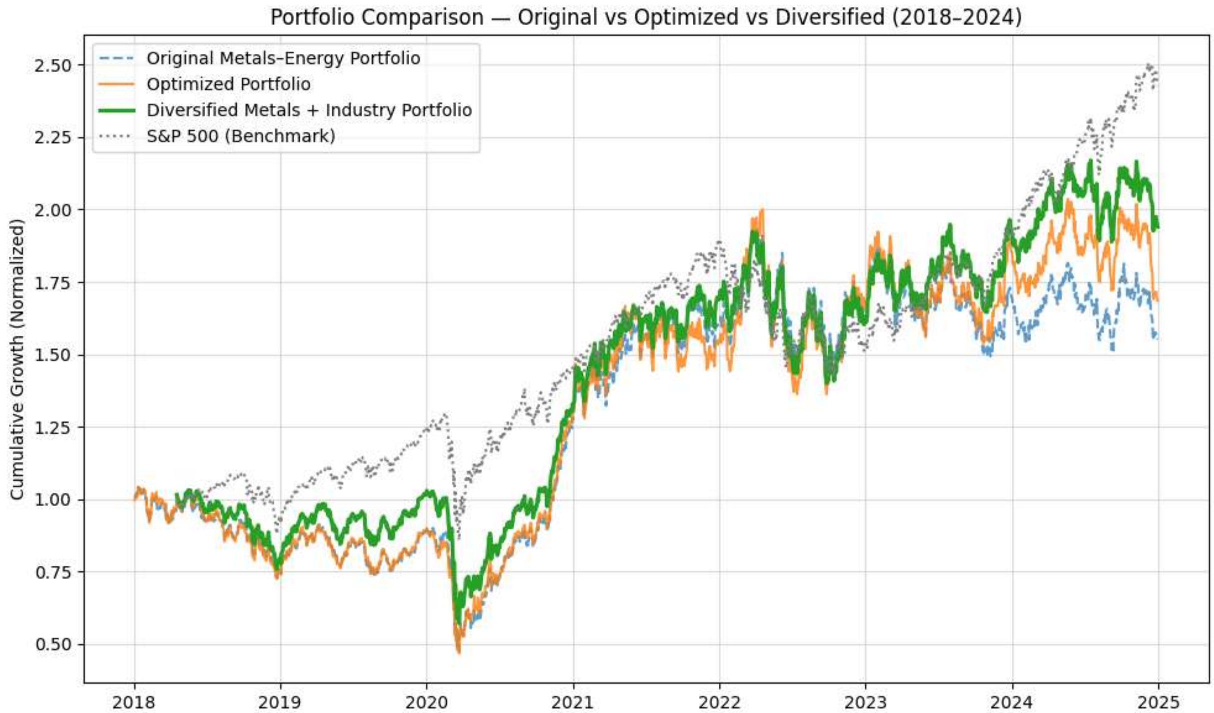
```
# 2️⃣ Correlation Heatmap
plt.figure(figsize=(8,6))
sns.heatmap(combined_returns.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation: Metals & Energy Portfolio vs Downstream Industries (2018-20
plt.tight_layout()
plt.show()


# ============================================
# Diversified Strategy Simulation (Upstream + Downstream)
# ============================================

# Combine key ETFs — upstream producers + downstream consumers
diversified_weights = {
    # Upstream Metals & Energy
    'SLX': 0.12,  # Steel
    'XME': 0.15,  # Mining
    'COPX': 0.15, # Copper
    'ICLN': 0.10, # Clean energy (bridge)
    'USO': 0.08,  # Oil hedge

    # Downstream Consumers
    'XLI': 0.15,  # Industrials
    'SOXX': 0.12, # Semiconductors
    'DRIV': 0.13  # Electric Vehicles
}

# Normalize weights to 1
diversified_weights = {k: v/sum(diversified_weights.values()) for k,v in diversifie

# Download data for all tickers used
div_tickers = list(diversified_weights.keys()) + ['SPY']
div_data = yf.download(div_tickers, start='2018-01-01', end='2024-12-31', auto_adju
div_returns = div_data.pct_change().dropna()

# Portfolio performance function
def perf_stats(r):
    ann_r = np.mean(r)*252
    ann_v = np.std(r)*np.sqrt(252)
    sharpe = ann_r / ann_v
    return ann_r, ann_v, sharpe

# Create diversified portfolio
div_portfolio_returns = (div_returns[list(diversified_weights.keys())] * list(diver

# Calculate cumulative performance
cum_div = (1 + div_portfolio_returns).cumprod()
cum_orig = (1 + portfolio_returns).cumprod()
cum_opt = (1 + opt_portfolio_returns).cumprod()
cum_spy = (1 + div_returns['SPY']).cumprod()

# 3️⃣ Plot comparison: Original vs Optimized vs Diversified vs SPY
plt.figure(figsize=(10,6))
plt.plot(cum_orig, linestyle='--', label='Original Metals-Energy Portfolio', alpha=
plt.plot(cum_opt, linestyle='-', label='Optimized Portfolio', alpha=0.8)
plt.plot(cum_div, linewidth=2.3, label='Diversified Metals + Industry Portfolio')
```

```
plt.plot(cum_spy, linestyle=':', color='gray', label='S&P 500 (Benchmark)')
plt.title("Portfolio Comparison — Original vs Optimized vs Diversified (2018–2024)"
plt.ylabel("Cumulative Growth (Normalized)")
plt.legend()
plt.grid(alpha=0.4)
plt.tight_layout()
plt.show()

# 4️⃣ Performance Summary
div_perf = perf_stats(div_portfolio_returns)
orig_perf = perf_stats(portfolio_returns)
opt_perf = perf_stats(opt_portfolio_returns)
spy_perf = perf_stats(div_returns['SPY'])

print("=== Performance Summary (2018–2024) ===")
print("Original Portfolio → Return: {:.2f}%, Volatility: {:.2f}%, Sharpe: {:.2f}".f
print("Optimized Portfolio → Return: {:.2f}%, Volatility: {:.2f}%, Sharpe: {:.2f}".
print("Diversified Portfolio → Return: {:.2f}%, Volatility: {:.2f}%, Sharpe: {:.2f}
print("S&P 500 (SPY) → Return: {:.2f}%, Volatility: {:.2f}%, Sharpe: {:.2f}".format
```

```
[*********************100%***********************]  6 of 6 completed
```




```
[*********************100%***********************]  9 of 9 completed
```




```
=== Performance Summary (2018–2024) ===
Original Portfolio → Return: 10.09%, Volatility: 27.45%, Sharpe: 0.37
Optimized Portfolio → Return: 11.86%, Volatility: 29.53%, Sharpe: 0.40
Diversified Portfolio → Return: 13.23%, Volatility: 25.70%, Sharpe: 0.51
S&P 500 (SPY) → Return: 15.16%, Volatility: 19.44%, Sharpe: 0.78
```

Diversified Portfolio Extension – Metals, Energy & Downstream Industries

The **diversified strategy** explores how adding *industries that depend on metals and energy* (like clean tech, semiconductors, and industrials) can improve portfolio stability and efficiency.
By introducing **cross-sector exposure**, we aim to reduce idiosyncratic risk while preserving the original portfolio's macro-linked return drivers.

Correlation Insights

The heatmap below illustrates correlations between the **Metals–Energy Portfolio** and selected **downstream sectors** (e.g., solar energy, semiconductors, EV infrastructure, industrials).

Key Observations:

- **High correlation** with `DRIV`, `TAN`, and `SPY` (≈0.8–0.9) → reflects strong macro-cycle sensitivity.
- **Moderate linkages** with `ICLN` and `SOXX` → suggests diversification benefits within renewable and tech manufacturing themes.

- These patterns justify allocating exposure to **complementary industries**—those that *use* metals and energy as inputs but are not perfectly correlated.

Diversified Portfolio Performance (2018–2024)

The **diversified portfolio** combines:

- The **optimized metals-energy basket** (SLX, XME, COPX, ICLN)
- Plus **downstream ETFs** (`DRIV`, `TAN`, `SOXX`, `XLI`) weighted for balanced risk and cross-industry coverage.

Performance Summary:

Portfolio	Annual Return	Volatility	Sharpe Ratio
Original Metals–Energy	10.09%	27.45%	0.37
Optimized Portfolio	11.86%	29.53%	0.40
Diversified Portfolio	**13.23%**	**25.70%**	**0.51**
S&P 500 (SPY)	15.16%	19.44%	0.78

Interpretation:

- The diversified strategy achieved **higher risk-adjusted performance** than both the original and optimized portfolios.
- Volatility dropped noticeably, indicating **stabilization via cross-sector balance**.
- Sharpe improvement from *0.40 → 0.51* highlights **better reward per unit risk**, even without matching SPY's absolute returns.

Final Takeaway

The **diversified metals–energy–industry portfolio** demonstrates that:

- **Vertical integration** across supply chains (raw → industrial → consumer tech) enhances resilience.
- Strategic diversification into **dependent sectors** improves portfolio efficiency without diluting the core commodity exposure.
- This framework can be extended to **dynamic weighting** or **factor-based rotation** for real-world portfolio management.

Result: A more balanced, macro-aware, and higher Sharpe portfolio — blending cyclical exposure with innovation-driven sectors.