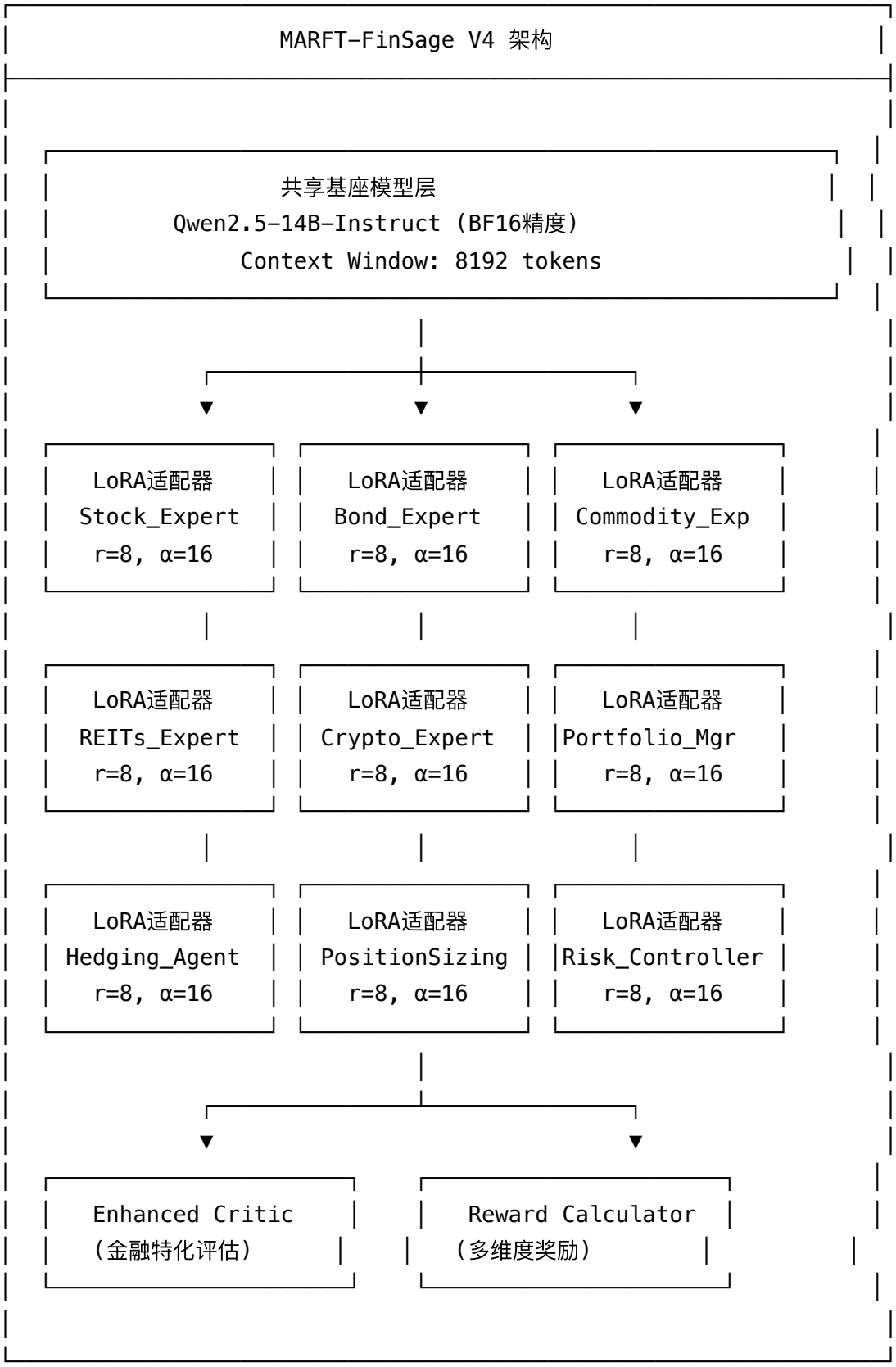
MARFT-FinSage V4 框架技术研究报告

版本: V4.0
日期: 2024-12-16
作者: FinSage Research Team

目录

1. 系统架构概述


MARFT-FinSage V4 架构
共享基座模型层
Qwen2.5-14B-Instruct (BF16精度)
Context Window: 8192 tokens
LoRA适配器 Stock_Expert r=8, α=16
LoRA适配器 Bond_Expert r=8, α=16
LoRA适配器 Commodity_Exp r=8, α=16
LoRA适配器 REITs_Expert r=8, α=16
LoRA适配器 Crypto_Expert r=8, α=16
LoRA适配器 Portfolio_Mgr r=8, α=16
LoRA适配器 Hedging_Agent r=8, α=16
LoRA适配器 PositionSizing r=8, α=16
LoRA适配器 Risk_Controller r=8, α=16
Enhanced Critic (金融特化评估)
Reward Calculator (多维度奖励)

2. 智能体配置详解

2.1 资产类别专家 (5个)

专家名称	资产类别	管理资产	依赖关系
Stock_Expert	stocks	SPY, QQQ, VTI, IWM, EFA	无 (独立决策)
Bond_Expert	bonds	TLT, IEF, BND, LQD, HYG	Stock_Expert
Commodity_Expert	commodities	GLD, SLV, USO, DBC, UNG	Stock_Expert, Bond_Expert
REITs_Expert	reits	VNQ, SCHH, IYR, REM, MORT	Stock_Expert, Bond_Expert
Crypto_Expert	crypto	BTC, ETH	Stock_Expert

2.2 元级别智能体 (4个)

智能体名称	功能	依赖关系
Portfolio_Manager	整合专家建议, 生成最终配置	所有5个资产专家
Position_Sizing_Agent	风险平价+Kelly准则仓位优化	Portfolio_Manager
Hedging_Agent	尾部风险对冲策略	Portfolio_Manager, Position_Sizing_Agent
Risk_Controller	最终风险审批与合规检查	所有智能体

2.3 依赖链详解

```
Level 0 (独立):
└── Stock_Expert (无依赖，首先执行)

Level 1 (单依赖):
├── Bond_Expert ← Stock_Expert
└── Crypto_Expert ← Stock_Expert

Level 2 (双依赖):
├── Commodity_Expert ← Stock_Expert, Bond_Expert
└── REITs_Expert ← Stock_Expert, Bond_Expert

Level 3 (全依赖):
└── Portfolio_Manager ← 所有5个资产专家

Level 4 (顺序依赖):
└── Position_Sizing_Agent ← Portfolio_Manager

Level 5:
└── Hedging_Agent ← Portfolio_Manager, Position_Sizing_Agent

Level 6 (最终门控):
└── Risk_Controller ← 所有智能体
```

3. LoRA微调配置

```
LORA_CONFIG = {
    "r": 8,                    # LoRA秩（低秩分解维度）
    "lora_alpha": 16,          # 缩放因子（通常设为2*r）
    "target_modules": [
        "q_proj",              # Query投影
        "v_proj",              # Value投影
        "k_proj",              # Key投影
        "o_proj",              # Output投影
    ],
    "lora_dropout": 0.05,      # Dropout比率
    "bias": "none",            # 不训练bias
    "task_type": "CAUSAL_LM",  # 因果语言模型任务
}
```

3.1 参数量分析

组件	参数量
基座模型 (Qwen2.5-14B)	~14B (冻结)
单个LoRA适配器	~2.6M (可训练)
9个LoRA适配器总计	~23.4M
可训练参数占比	**~0.17%**

4. 奖励函数设计

4.1 ExpertReward (资产专家奖励)

```
class ExpertReward:
    """资产类别专家的奖励函数"""

    # 权重配置
    accuracy_weight: float = 0.35      # 预测准确性
    calibration_weight: float = 0.20   # 置信度校准
    timing_weight: float = 0.25        # 时机把握
    contribution_weight: float = 0.20  # 组合贡献度

    def compute(self, expert_action, market_outcome, portfolio_context):
        # 1. 方向准确性奖励
        direction_reward = self._compute_direction_accuracy(
            predicted=expert_action["direction"],
            actual=market_outcome["return"]
        )

        # 2. 置信度校准奖励 (Brier Score)
        calibration_reward = self._compute_calibration(
            confidence=expert_action["confidence"],
            was_correct=direction_reward > 0
        )

        # 3. 时机奖励 (是否在正确时间行动)
        timing_reward = self._compute_timing(
            action_time=expert_action["timestamp"],
            optimal_time=market_outcome["optimal_entry"]
        )

        # 4. 组合贡献奖励
        contribution_reward = self._compute_contribution(
            expert_return=expert_action["asset_return"],
            portfolio_return=portfolio_context["total_return"]
        )

        total = (
            self.accuracy_weight * direction_reward +
            self.calibration_weight * calibration_reward +
            self.timing_weight * timing_reward +
```

```
        self.contribution_weight * contribution_reward
    )

    return total
```

4.2 PortfolioManagerReward (组合管理奖励)

```python
class PortfolioManagerReward:
    """投资组合管理器奖励"""

    return_weight: float = 0.35         # 收益贡献
    consensus_weight: float = 0.25      # 专家共识整合
    quality_weight: float = 0.25        # 决策质量
    timing_weight: float = 0.15         # 再平衡时机

    def compute(self, allocation, expert_opinions, market_outcome):
        # 1. 风险调整收益
        return_reward = self._risk_adjusted_return(
            portfolio_return=market_outcome["portfolio_return"],
            volatility=market_outcome["realized_vol"]
        )

        # 2. 专家共识整合度
        consensus_reward = self._consensus_integration(
            final_allocation=allocation,
            expert_recommendations=expert_opinions,
            expert_confidences=[e["confidence"] for e in expert_opinions]
        )

        # 3. 决策质量（夏普比率改善）
        quality_reward = self._decision_quality(
            sharpe_before=market_outcome["sharpe_before"],
            sharpe_after=market_outcome["sharpe_after"]
        )

        # 4. 再平衡时机
        timing_reward = self._rebalance_timing(
            rebalance_cost=market_outcome["transaction_cost"],
            rebalance_benefit=market_outcome["tracking_improvement"]
        )

        return (
            self.return_weight * return_reward +
            self.consensus_weight * consensus_reward +
            self.quality_weight * quality_reward +
            self.timing_weight * timing_reward
        )
```

4.3 PositionSizingReward (仓位管理奖励)

```python
class PositionSizingReward:
    """仓位管理智能体奖励"""

    risk_parity_weight: float = 0.30   # 风险平价达成度
    kelly_weight: float = 0.25         # Kelly准则遵循度
    vol_target_weight: float = 0.30    # 波动率目标达成
    liquidity_weight: float = 0.15     # 流动性考量

    def compute(self, position_sizes, risk_metrics, market_conditions):
        # 1. 风险平价得分
        risk_parity_score = self._compute_risk_parity(
            weights=position_sizes,
            covariance=risk_metrics["covariance_matrix"]
        )

        # 2. Kelly准则遵循度
        kelly_score = self._compute_kelly_adherence(
            actual_sizes=position_sizes,
            kelly_optimal=risk_metrics["kelly_fractions"]
        )

        # 3. 波动率目标达成
        vol_score = self._compute_vol_targeting(
            realized_vol=risk_metrics["realized_vol"],
            target_vol=risk_metrics["target_vol"]
        )

        # 4. 流动性调整
        liquidity_score = self._compute_liquidity_adjustment(
            position_sizes=position_sizes,
            liquidity_scores=market_conditions["liquidity"]
        )

        return (
            self.risk_parity_weight * risk_parity_score +
            self.kelly_weight * kelly_score +
            self.vol_target_weight * vol_score +
            self.liquidity_weight * liquidity_score
        )
```

4.4 HedgingReward (对冲策略奖励)

```python
class HedgingReward:
    """对冲智能体奖励"""

    tail_risk_weight: float = 0.35      # 尾部风险保护
    cost_efficiency_weight: float = 0.25  # 成本效率
    vix_response_weight: float = 0.25  # VIX响应
    dynamic_weight: float = 0.15       # 动态调整能力

    def compute(self, hedge_action, risk_event, cost_metrics):
        # 1. 尾部风险保护效果
        tail_protection = self._compute_tail_protection(
            portfolio_drawdown=risk_event["max_drawdown"],
            hedged_drawdown=risk_event["hedged_drawdown"]
        )

        # 2. 对冲成本效率
        cost_efficiency = self._compute_cost_efficiency(
            hedge_cost=cost_metrics["premium_paid"],
            protection_value=cost_metrics["payout_received"]
        )

        # 3. VIX响应及时性
        vix_response = self._compute_vix_response(
            vix_spike=risk_event["vix_change"],
            hedge_adjustment_speed=hedge_action["adjustment_lag"]
        )

        # 4. 动态调整得分
        dynamic_score = self._compute_dynamic_adjustment(
            hedge_ratio_changes=hedge_action["ratio_adjustments"],
            market_regime_changes=risk_event["regime_shifts"]
        )

        return (
            self.tail_risk_weight * tail_protection +
            self.cost_efficiency_weight * cost_efficiency +
            self.vix_response_weight * vix_response +
            self.dynamic_weight * dynamic_score
        )
```

4.5 CoordinationReward (协调奖励)

```python
class CoordinationReward:
    """多智能体协调奖励"""

    consistency_weight: float = 0.25        # 信息一致性
    info_utilization_weight: float = 0.25  # 信息利用率
    conflict_resolution_weight: float = 0.25  # 冲突解决
    efficiency_weight: float = 0.25         # 协作效率

    def compute(self, agent_outputs, dependency_graph):
        # 1. 信息一致性
        consistency = self._compute_consistency(
            upstream_signals=[a["signal"] for a in agent_outputs["predecessors"]],
            downstream_action=agent_outputs["current"]["action"]
        )

        # 2. 信息利用率
        utilization = self._compute_info_utilization(
            available_info=agent_outputs["predecessors"],
            used_info=agent_outputs["current"]["cited_sources"]
        )

        # 3. 冲突解决质量
        conflict_resolution = self._compute_conflict_resolution(
            conflicting_signals=agent_outputs["conflicts"],
            resolution_quality=agent_outputs["resolution_score"]
        )

        # 4. 协作效率
        efficiency = self._compute_efficiency(
            decision_latency=agent_outputs["latency"],
            decision_quality=agent_outputs["quality_score"]
        )

        return (
            self.consistency_weight * consistency +
            self.info_utilization_weight * utilization +
            self.conflict_resolution_weight * conflict_resolution +
            self.efficiency_weight * efficiency
        )
```

4.6 CombinedRewardCalculator (综合奖励)

```
class CombinedRewardCalculator:
    """综合奖励计算器"""

    individual_weight: float = 0.4    # 个体表现
    team_weight: float = 0.4          # 团队表现
    coordination_weight: float = 0.2  # 协调奖励

    def compute_total_reward(self, agent_id, individual_reward, team_reward, coord_rewa
        return (
            self.individual_weight * individual_reward +
            self.team_weight * team_reward +
            self.coordination_weight * coord_reward
        )
```

5. PPO训练配置

```
PPO_CONFIG = {
    # 核心超参数
    "clip_param": 0.2,              # PPO裁剪参数
    "gamma": 0.99,                  # 折扣因子
    "gae_lambda": 0.95,             # GAE参数
    "ppo_epochs": 4,                # 每次更新的epoch数

    # 学习率
    "policy_lr": 5e-6,              # 策略网络学习率（LLM需要很小的lr）
    "critic_lr": 1e-4,              # Critic学习率

    # 批次设置
    "rollout_length": 20,           # 轨迹长度
    "num_mini_batches": 4,          # Mini-batch数量

    # 正则化
    "entropy_coef": 0.01,           # 熵正则化系数
    "value_loss_coef": 0.5,         # Value loss系数
    "max_grad_norm": 0.5,           # 梯度裁剪

    # 训练控制
    "kl_threshold": 0.01,           # KL散度阈值
    "early_stop_kl": 0.02,          # 早停KL阈值
}
```

5.1 GAE计算

```python
def compute_gae(rewards, values, next_value, gamma=0.99, gae_lambda=0.95):
    """计算广义优势估计 (GAE)"""
    advantages = []
    gae = 0

    for t in reversed(range(len(rewards))):
        if t == len(rewards) - 1:
            next_val = next_value
        else:
            next_val = values[t + 1]

        delta = rewards[t] + gamma * next_val - values[t]
        gae = delta + gamma * gae_lambda * gae
        advantages.insert(0, gae)

    return advantages
```

6. Critic网络架构

```
class EnhancedCritic(nn.Module):
    """增强型Critic网络"""

    def __init__(
        self,
        num_assets: int = 50,
        hidden_size: int = 512,
        num_agents: int = 5,
        num_layers: int = 3
    ):
        super().__init__()

        # 输入维度计算
        # 资产特征: num_assets * 8 (价格、收益、波动率等)
        # 宏观特征: 20 (利率、VIX、经济指标等)
        # 组合特征: num_assets (当前权重)
        # 智能体特征: num_agents * 13 (每个智能体的动作one-hot)
        input_dim = num_assets * 8 + 20 + num_assets + num_agents * 13

        # 特征处理层
        self.asset_encoder = nn.Sequential(
            nn.Linear(num_assets * 8, hidden_size),
            nn.LayerNorm(hidden_size),
            nn.GELU()
        )

        self.macro_encoder = nn.Sequential(
            nn.Linear(20, hidden_size // 4),
            nn.LayerNorm(hidden_size // 4),
            nn.GELU()
        )

        # Transformer编码器
        self.transformer = nn.TransformerEncoder(
            nn.TransformerEncoderLayer(
                d_model=hidden_size,
                nhead=8,
                dim_feedforward=hidden_size * 4,
                dropout=0.1,
                batch_first=True
```

```
    ),
    num_layers=num_layers
)

# 价值头
self.value_head = nn.Sequential(
    nn.Linear(hidden_size, hidden_size // 2),
    nn.GELU(),
    nn.Linear(hidden_size // 2, 1)
)
```

7. 动作空间设计

7.1 TradeAction枚举

```
class TradeAction(Enum):
    """交易动作空间 (13个离散动作)"""

    # 做空动作 (4个)
    SHORT_100 = 0    # 做空100%
    SHORT_50 = 1     # 做空50%
    SHORT_25 = 2     # 做空25%
    SHORT_10 = 3     # 做空10%

    # 卖出动作 (2个)
    SELL_50 = 4      # 卖出50%
    SELL_25 = 5      # 卖出25%

    # 持有
    HOLD = 6         # 维持现状

    # 买入动作 (4个)
    BUY_10 = 7       # 买入10%
    BUY_25 = 8       # 买入25%
    BUY_50 = 9       # 买入50%
    BUY_100 = 10     # 买入100%

    # 强信号动作 (2个)
    STRONG_BUY = 11  # 强烈买入
    STRONG_SELL = 12 # 强烈卖出
```

7.2 动作到权重映射

```
ACTION_TO_WEIGHT_DELTA = {
    TradeAction.SHORT_100: -1.00,
    TradeAction.SHORT_50: -0.50,
    TradeAction.SHORT_25: -0.25,
    TradeAction.SHORT_10: -0.10,
    TradeAction.SELL_50: -0.50,
    TradeAction.SELL_25: -0.25,
    TradeAction.HOLD: 0.00,
    TradeAction.BUY_10: +0.10,
    TradeAction.BUY_25: +0.25,
    TradeAction.BUY_50: +0.50,
    TradeAction.BUY_100: +1.00,
    TradeAction.STRONG_BUY: +1.50,
    TradeAction.STRONG_SELL: -1.50,
}
```

8. 辅助模块集成

8.1 因子评分器 (5个)

因子名称	描述	评分范围
MomentumFactorScorer	动量因子评分	[-1, 1]
ValueFactorScorer	价值因子评分	[-1, 1]
QualityFactorScorer	质量因子评分	[-1, 1]
VolatilityFactorScorer	波动率因子评分	[-1, 1]
SizeFactorScorer	规模因子评分	[-1, 1]

8.2 对冲工具 (11个)

工具名称	功能
MinimumVarianceHedge	最小方差对冲

工具名称	功能
RiskParityHedge	风险平价对冲
MaxSharpeHedge	最大夏普比率对冲
CVaROptimizationHedge	CVaR优化对冲
DCCGARCHHedge	DCC-GARCH动态对冲
HRPHedge	层次风险平价
TailRiskHedge	尾部风险对冲
VolatilityTargetingHedge	波动率目标对冲
CorrelationHedge	相关性对冲
FactorHedge	因子对冲
DynamicHedge	动态对冲

8.3 策略类 (6个)

策略名称	描述
StrategicAllocationStrategy	战略资产配置
TacticalAllocationStrategy	战术资产配置
CoreSatelliteStrategy	核心-卫星策略
DynamicRebalancingStrategy	动态再平衡策略
MomentumStrategy	动量策略
MeanReversionStrategy	均值回归策略

9. 推理加速配置

```
INFERENCE_CONFIG = {
    # vLLM配置
    "use_vllm": True,
    "tensor_parallel_size": 1,       # 单GPU
    "max_num_seqs": 9,               # 9个智能体并行
    "max_model_len": 8192,           # 最大序列长度

    # KV Cache配置
    "enable_kv_cache": True,
    "kv_cache_dtype": "auto",

    # 批处理配置
    "batch_size": 9,                 # 9个智能体同时推理
    "dynamic_batching": True,

    # 量化配置（可选）
    "quantization": None,            # 使用BF16全精度
}
```

10. 环境配置

```
ENV_CONFIG = {
    # 初始资本
    "initial_capital": 1_000_000,

    # 交易成本
    "transaction_cost": 0.001,      # 0.1%
    "slippage": 0.0005,             # 0.05%

    # 约束条件
    "max_single_weight": 0.15,      # 单资产最大15%
    "max_class_weight": 0.50,       # 单类别最大50%
    "rebalance_threshold": 0.02,    # 2%偏离触发再平衡

    # 数据范围
    "start_date": "2020-01-01",
    "end_date": "2024-12-31",
    "frequency": "daily",

    # 评估指标
    "risk_free_rate": 0.04,         # 无风险利率
    "benchmark": "SPY",             # 基准
}
```

11. 智能体输入输出示例

11.1 Stock_Expert (股票专家)

输入示例：

```
{
  "observation": {
    "date": "2024-03-15",
    "market_data": {
      "SPY": {"price": 512.45, "change_1d": 0.0125, "change_5d": 0.0287, "volume_ratio"
      "QQQ": {"price": 438.72, "change_1d": 0.0156, "change_5d": 0.0342, "volume_ratio"
      "VTI": {"price": 256.33, "change_1d": 0.0118, "change_5d": 0.0265, "volume_ratio"
    },
    "technical_indicators": {
      "SPY": {"rsi_14": 62.5, "macd_signal": 0.85, "bb_position": 0.72, "sma_20_cross":
      "QQQ": {"rsi_14": 68.3, "macd_signal": 1.12, "bb_position": 0.81, "sma_20_cross":
    },
    "macro_context": {
      "vix": 14.25,
      "fed_funds_rate": 5.25,
      "yield_10y": 4.28,
      "economic_surprise_index": 32.5
    },
    "sector_momentum": {
      "technology": 0.045,
      "healthcare": 0.012,
      "financials": 0.028,
      "energy": -0.015
    }
  },
  "predecessor_actions": null,
  "current_position": {"SPY": 0.08, "QQQ": 0.05, "VTI": 0.03}
}
```


输出示例：

```
{
  "action": "BUY_25%",
  "action_id": 10,
  "confidence": 0.78,
  "target_allocation": {
    "SPY": 0.10,
    "QQQ": 0.06,
    "VTI": 0.04
  },
  "reasoning": "技术面偏强：SPY RSI=62.5处于健康区间，MACD看涨信号明确。宏观环境支持：VIX=14.25处
  "key_factors": [
    {"factor": "momentum", "score": 0.82, "weight": 0.3},
    {"factor": "volatility_regime", "score": 0.75, "weight": 0.25},
    {"factor": "macro_support", "score": 0.71, "weight": 0.25},
    {"factor": "sector_rotation", "score": 0.68, "weight": 0.2}
  ]
}
```


11.2 Bond_Expert (债券专家)

输入示例：

```
{
  "observation": {
    "date": "2024-03-15",
    "market_data": {
      "TLT": {"price": 92.45, "change_1d": -0.0045, "duration": 17.2, "yield": 4.52},
      "IEF": {"price": 94.82, "change_1d": -0.0028, "duration": 7.5, "yield": 4.38},
      "BND": {"price": 72.15, "change_1d": -0.0032, "duration": 6.1, "yield": 4.85}
    },
    "yield_curve": {
      "2y": 4.65,
      "5y": 4.32,
      "10y": 4.28,
      "30y": 4.45,
      "2y10y_spread": -0.37,
      "curve_slope": "inverted"
    },
    "credit_spreads": {
      "ig_spread": 95,
      "hy_spread": 345,
      "ig_spread_change": -5,
      "hy_spread_change": -12
    },
    "inflation_expectations": {
      "breakeven_5y": 2.35,
      "breakeven_10y": 2.28,
      "tips_real_yield": 1.95
    }
  },
  "predecessor_actions": {
    "Stock_Expert": {
      "action": "BUY_25%",
      "confidence": 0.78,
      "reasoning": "股票偏多，风险偏好上升"
    }
  },
  "current_position": {"TLT": 0.05, "IEF": 0.08, "BND": 0.07}
}
```


输出示例：

```
{
  "action": "HOLD",
  "action_id": 6,
  "confidence": 0.65,
  "target_allocation": {
    "TLT": 0.04,
    "IEF": 0.09,
    "BND": 0.07
  },
  "reasoning": "收益率曲线倒挂(-37bp)暗示经济放缓风险，但股票专家看涨(置信度78%)表明短期风险偏好改善
  "duration_recommendation": {
    "target_duration": 7.8,
    "current_duration": 8.5,
    "adjustment": "reduce_long_end"
  },
  "key_factors": [
    {"factor": "yield_curve_signal", "score": 0.45, "weight": 0.3},
    {"factor": "credit_environment", "score": 0.72, "weight": 0.25},
    {"factor": "stock_expert_alignment", "score": 0.68, "weight": 0.25},
    {"factor": "inflation_outlook", "score": 0.62, "weight": 0.2}
  ]
}
```


11.3 Commodity_Expert (大宗商品专家)

输入示例：

```
{
  "observation": {
    "date": "2024-03-15",
    "market_data": {
      "GLD": {"price": 198.75, "change_1d": 0.0082, "change_20d": 0.0345},
      "USO": {"price": 78.42, "change_1d": -0.0125, "change_20d": 0.0156},
      "DBC": {"price": 23.15, "change_1d": 0.0045, "change_20d": 0.0212}
    },
    "commodity_fundamentals": {
      "gold": {
        "real_rate_correlation": -0.72,
        "central_bank_buying": "strong",
        "etf_flows_20d": 125000000
      },
      "oil": {
        "inventory_change": -2500000,
        "opec_compliance": 0.92,
        "demand_forecast_revision": 0.5
      }
    },
    "currency_impact": {
      "dxy": 103.45,
      "dxy_change_20d": -0.012,
      "usd_correlation": -0.65
    },
    "geopolitical_risk_index": 145
  },
  "predecessor_actions": {
    "Stock_Expert": {"action": "BUY_25%", "confidence": 0.78},
    "Bond_Expert": {"action": "HOLD", "confidence": 0.65}
  },
  "current_position": {"GLD": 0.06, "USO": 0.02, "DBC": 0.02}
}
```


输出示例：

```
{
  "action": "BUY_10%",
  "action_id": 9,
  "confidence": 0.72,
  "target_allocation": {
    "GLD": 0.08,
    "USO": 0.02,
    "DBC": 0.02
  },
  "reasoning": "黄金配置价值凸显：实际利率与金价负相关(-0.72)，美元走弱(DXY -1.2%)支持金价。央行持
  "commodity_view": {
    "gold": "overweight",
    "oil": "neutral",
    "broad_commodities": "neutral"
  },
  "key_factors": [
    {"factor": "real_rate_environment", "score": 0.78, "weight": 0.3},
    {"factor": "usd_weakness", "score": 0.71, "weight": 0.25},
    {"factor": "geopolitical_hedge", "score": 0.75, "weight": 0.25},
    {"factor": "portfolio_diversification", "score": 0.68, "weight": 0.2}
  ]
}
```


11.4 REITs_Expert (房地产信托专家)

输入示例：

```
{
  "observation": {
    "date": "2024-03-15",
    "market_data": {
      "VNQ": {"price": 85.32, "change_1d": 0.0065, "dividend_yield": 4.25, "nav_premium
      "SCHH": {"price": 19.45, "change_1d": 0.0058, "dividend_yield": 3.95, "nav_premiu
    },
    "sector_breakdown": {
      "residential": {"weight": 0.15, "occupancy": 0.945, "rent_growth": 0.035},
      "industrial": {"weight": 0.12, "occupancy": 0.972, "rent_growth": 0.065},
      "office": {"weight": 0.08, "occupancy": 0.875, "rent_growth": -0.015},
      "retail": {"weight": 0.10, "occupancy": 0.915, "rent_growth": 0.022},
      "data_center": {"weight": 0.08, "occupancy": 0.985, "rent_growth": 0.082}
    },
    "interest_rate_sensitivity": {
      "rate_beta": -0.45,
      "spread_to_10y": 0.25,
      "historical_avg_spread": 0.15
    },
    "valuation_metrics": {
      "ffo_yield": 5.85,
      "price_to_nav": 0.92,
      "implied_cap_rate": 5.65
    }
  },
  "predecessor_actions": {
    "Stock_Expert": {"action": "BUY_25%", "confidence": 0.78},
    "Bond_Expert": {"action": "HOLD", "confidence": 0.65}
  },
  "current_position": {"VNQ": 0.04, "SCHH": 0.02}
}
```


输出示例：

```
{
  "action": "BUY_10%",
  "action_id": 9,
  "confidence": 0.68,
  "target_allocation": {
    "VNQ": 0.05,
    "SCHH": 0.02
  },
  "reasoning": "REITs估值具吸引力：NAV折价8%，FFO收益率5.85%高于历史均值。板块分化明显：数据中心(+
  "sector_preference": {
    "overweight": ["data_center", "industrial"],
    "neutral": ["residential", "retail"],
    "underweight": ["office"]
  },
  "key_factors": [
    {"factor": "valuation_discount", "score": 0.75, "weight": 0.3},
    {"factor": "sector_fundamentals", "score": 0.72, "weight": 0.25},
    {"factor": "rate_sensitivity", "score": 0.62, "weight": 0.25},
    {"factor": "diversification_benefit", "score": 0.70, "weight": 0.2}
  ]
}
```


11.5 Crypto_Expert (加密货币专家)

输入示例：

```
{
  "observation": {
    "date": "2024-03-15",
    "market_data": {
      "BTC": {"price": 72450, "change_1d": 0.0285, "change_7d": 0.0856, "market_cap": 1
      "ETH": {"price": 4025, "change_1d": 0.0312, "change_7d": 0.0725, "market_cap": 48
    },
    "on_chain_metrics": {
      "btc": {
        "active_addresses_7d": 925000,
        "exchange_netflow_7d": -15000,
        "mvrv_ratio": 2.15,
        "nupl": 0.58
      },
      "eth": {
        "active_addresses_7d": 485000,
        "staking_rate": 0.265,
        "gas_fees_avg": 45
      }
    },
    "market_structure": {
      "btc_dominance": 0.525,
      "total_market_cap": 2700000000000,
      "fear_greed_index": 72,
      "funding_rate": 0.015
    },
    "institutional_flow": {
      "etf_flow_7d": 850000000,
      "grayscale_premium": -0.02
    }
  },
  "predecessor_actions": {
    "Stock_Expert": {"action": "BUY_25%", "confidence": 0.78}
  },
  "current_position": {"BTC": 0.02, "ETH": 0.01}
}
```


输出示例：

```
{
  "action": "BUY_10%",
  "action_id": 9,
  "confidence": 0.70,
  "target_allocation": {
    "BTC": 0.025,
    "ETH": 0.015
  },
  "reasoning": "链上数据支持牛市：交易所净流出15000 BTC，供应紧缩。MVRV=2.15处于合理区间(非泡沫),
  "risk_assessment": {
    "volatility_regime": "elevated",
    "max_recommended_allocation": 0.05,
    "correlation_to_stocks": 0.62
  },
  "key_factors": [
    {"factor": "on_chain_health", "score": 0.76, "weight": 0.3},
    {"factor": "institutional_demand", "score": 0.82, "weight": 0.25},
    {"factor": "market_sentiment", "score": 0.68, "weight": 0.25},
    {"factor": "stock_correlation", "score": 0.65, "weight": 0.2}
  ]
}
```


11.6 Portfolio_Manager (投资组合管理器)

输入示例：

```
{
  "observation": {
    "date": "2024-03-15",
    "portfolio_state": {
      "total_value": 1085000,
      "cash": 0.35,
      "positions": {
        "stocks": 0.20,
        "bonds": 0.20,
        "commodities": 0.10,
        "reits": 0.06,
        "crypto": 0.03
      }
    },
    "performance_metrics": {
      "return_ytd": 0.085,
      "volatility_30d": 0.12,
      "sharpe_ratio": 1.45,
      "max_drawdown": -0.065
    },
    "risk_metrics": {
      "var_95": -0.018,
      "cvar_95": -0.025,
      "beta_to_spy": 0.55
    }
  },
  "predecessor_actions": {
    "Stock_Expert": {"action": "BUY_25%", "confidence": 0.78, "target_allocation": {"SP
    "Bond_Expert": {"action": "HOLD", "confidence": 0.65, "target_allocation": {"TLT":
    "Commodity_Expert": {"action": "BUY_10%", "confidence": 0.72, "target_allocation":
    "REITs_Expert": {"action": "BUY_10%", "confidence": 0.68, "target_allocation": {"VN
    "Crypto_Expert": {"action": "BUY_10%", "confidence": 0.70, "target_allocation": {"B
  }
}
```


输出示例：

```
{
  "action": "REBALANCE",
  "action_id": 7,
  "confidence": 0.82,
  "final_allocation": {
    "SPY": 0.10,
    "QQQ": 0.06,
    "VTI": 0.04,
    "TLT": 0.04,
    "IEF": 0.09,
    "BND": 0.07,
    "GLD": 0.07,
    "USO": 0.02,
    "DBC": 0.02,
    "VNQ": 0.05,
    "SCHH": 0.02,
    "BTC": 0.02,
    "ETH": 0.01,
    "CASH": 0.29
  },
  "allocation_by_class": {
    "stocks": 0.20,
    "bonds": 0.20,
    "commodities": 0.11,
    "reits": 0.07,
    "crypto": 0.03,
    "cash": 0.29
  },
  "reasoning": "综合5位专家建议：股票(+25%,0.78)、商品(+10%,0.72)、加密(+10%,0.70)、REITs(+10
  "expert_weight_used": {
    "Stock_Expert": 0.26,
    "Bond_Expert": 0.22,
    "Commodity_Expert": 0.20,
    "REITs_Expert": 0.18,
    "Crypto_Expert": 0.14
  },
  "rebalance_trades": [
    {"asset": "SPY", "action": "BUY", "amount": 21650, "shares": 42},
    {"asset": "GLD", "action": "BUY", "amount": 10850, "shares": 54},
    {"asset": "VNQ", "action": "BUY", "amount": 10850, "shares": 127}
  ]
}
```

11.7 Position_Sizing_Agent (仓位管理智能体)

输入示例：

```
{
  "observation": {
    "date": "2024-03-15",
    "portfolio_allocation": {
      "SPY": 0.10,
      "QQQ": 0.06,
      "TLT": 0.04,
      "GLD": 0.07
    },
    "volatility_estimates": {
      "SPY": {"vol_20d": 0.14, "vol_60d": 0.16, "garch_forecast": 0.15},
      "QQQ": {"vol_20d": 0.18, "vol_60d": 0.20, "garch_forecast": 0.19},
      "TLT": {"vol_20d": 0.12, "vol_60d": 0.14, "garch_forecast": 0.13},
      "GLD": {"vol_20d": 0.11, "vol_60d": 0.12, "garch_forecast": 0.11}
    },
    "correlation_matrix": {
      "SPY_QQQ": 0.92,
      "SPY_TLT": -0.35,
      "SPY_GLD": 0.15,
      "TLT_GLD": 0.28
    },
    "kelly_estimates": {
      "SPY": {"win_rate": 0.58, "avg_win": 0.012, "avg_loss": -0.008, "kelly_fraction":
      "QQQ": {"win_rate": 0.55, "avg_win": 0.015, "avg_loss": -0.011, "kelly_fraction":
    },
    "liquidity_scores": {
      "SPY": 1.0,
      "QQQ": 0.98,
      "TLT": 0.95,
      "GLD": 0.92
    }
  },
  "predecessor_actions": {
    "Portfolio_Manager": {
      "action": "REBALANCE",
      "confidence": 0.82,
      "final_allocation": {"SPY": 0.10, "QQQ": 0.06, "TLT": 0.04, "GLD": 0.07}
    }
  },
  "risk_budget": {
    "target_portfolio_vol": 0.12,
    "max_single_position": 0.15,
    "var_limit": 0.02
```

```
    }
  }
```

输出示例：

```
{
  "action": "ADJUST_SIZES",
  "action_id": 8,
  "confidence": 0.85,
  "adjusted_allocation": {
    "SPY": 0.095,
    "QQQ": 0.052,
    "TLT": 0.048,
    "GLD": 0.075
  },
  "sizing_rationale": {
    "SPY": {
      "pm_target": 0.10,
      "vol_adjusted": 0.098,
      "kelly_adjusted": 0.095,
      "final": 0.095,
      "reason": "略降以控制整体波动率"
    },
    "QQQ": {
      "pm_target": 0.06,
      "vol_adjusted": 0.055,
      "kelly_adjusted": 0.052,
      "final": 0.052,
      "reason": "高波动率(19%)需降低Kelly仓位"
    },
    "TLT": {
      "pm_target": 0.04,
      "vol_adjusted": 0.048,
      "kelly_adjusted": 0.048,
      "final": 0.048,
      "reason": "低波动率允许略增仓位，增强对冲"
    },
    "GLD": {
      "pm_target": 0.07,
      "vol_adjusted": 0.072,
      "kelly_adjusted": 0.075,
      "final": 0.075,
      "reason": "低相关性资产，风险平价支持增配"
    }
  },
  "portfolio_risk_metrics": {
    "expected_vol": 0.118,
    "contribution_to_risk": {
```

```
      "SPY": 0.42,
      "QQQ": 0.28,
      "TLT": 0.12,
      "GLD": 0.18
    },
    "diversification_ratio": 1.35
  },
  "key_factors": [
    {"factor": "risk_parity_score", "score": 0.82, "weight": 0.30},
    {"factor": "kelly_criterion", "score": 0.78, "weight": 0.25},
    {"factor": "vol_targeting", "score": 0.85, "weight": 0.30},
    {"factor": "liquidity_adjustment", "score": 0.92, "weight": 0.15}
  ]
}
```


11.8 Hedging_Agent (对冲智能体)

输入示例:

```
{
  "observation": {
    "date": "2024-03-15",
    "portfolio_exposure": {
      "equity_beta": 0.55,
      "duration": 4.2,
      "commodity_delta": 0.12,
      "total_value": 1085000
    },
    "volatility_surface": {
      "vix": 14.25,
      "vix_term_structure": {"1m": 14.25, "3m": 15.80, "6m": 17.20},
      "skew": -0.15,
      "put_call_ratio": 0.85
    },
    "tail_risk_indicators": {
      "skew_index": -128,
      "credit_spread_change_5d": -8,
      "em_stress_index": 45,
      "liquidity_index": 0.92
    },
    "hedging_instruments": {
      "SPY_put_atm": {"premium": 0.025, "delta": -0.50, "gamma": 0.08, "theta": -0.003}
      "VXX": {"price": 15.25, "beta_to_vix": 0.85},
      "TLT_call": {"premium": 0.018, "delta": 0.45}
    }
  },
  "predecessor_actions": {
    "Portfolio_Manager": {"action": "REBALANCE", "confidence": 0.82},
    "Position_Sizing_Agent": {"action": "ADJUST_SIZES", "confidence": 0.85}
  },
  "current_hedges": {
    "spy_puts": {"notional": 25000, "strike": 500, "expiry": "2024-04-19"},
    "vxx_long": {"notional": 5000}
  }
}
```


输出示例：

```
{
  "action": "ADJUST_HEDGE",
  "action_id": 8,
  "confidence": 0.75,
  "hedge_recommendations": {
    "spy_puts": {
      "current_notional": 25000,
      "recommended_notional": 20000,
      "action": "REDUCE",
      "reason": "VIX处于低位(14.25)，期权成本相对较高。期限结构contango，不急于对冲"
    },
    "vxx_long": {
      "current_notional": 5000,
      "recommended_notional": 3000,
      "action": "REDUCE",
      "reason": "contango造成持有成本，减少敞口"
    },
    "tlt_call": {
      "current_notional": 0,
      "recommended_notional": 8000,
      "action": "ADD",
      "reason": "增加利率对冲，债券久期提供股票下跌保护"
    }
  },
  "hedge_ratios": {
    "equity_hedge_ratio": 0.15,
    "rate_hedge_ratio": 0.20,
    "vol_hedge_ratio": 0.05
  },
  "cost_analysis": {
    "current_hedge_cost_annual": 0.008,
    "recommended_hedge_cost_annual": 0.006,
    "cost_savings": 0.002
  },
  "tail_risk_assessment": {
    "var_99_unhedged": -0.042,
    "var_99_hedged": -0.028,
    "protection_improvement": 0.33
  },
  "key_factors": [
    {"factor": "tail_risk_protection", "score": 0.72, "weight": 0.35},
    {"factor": "cost_efficiency", "score": 0.78, "weight": 0.25},
    {"factor": "vix_regime_response", "score": 0.65, "weight": 0.25},
```

```
    {"factor": "dynamic_adjustment", "score": 0.70, "weight": 0.15}
  ]
}
```

11.9 Risk_Controller (风险控制器)

输入示例:

```
{
  "observation": {
    "date": "2024-03-15",
    "final_proposed_portfolio": {
      "SPY": 0.095,
      "QQQ": 0.052,
      "VTI": 0.04,
      "TLT": 0.048,
      "IEF": 0.09,
      "BND": 0.07,
      "GLD": 0.075,
      "USO": 0.02,
      "VNQ": 0.05,
      "BTC": 0.02,
      "CASH": 0.35
    },
    "risk_metrics": {
      "portfolio_vol": 0.118,
      "var_95": -0.018,
      "cvar_95": -0.025,
      "max_drawdown_1y": -0.12,
      "sharpe_ratio": 1.45
    },
    "concentration_metrics": {
      "hhi": 0.082,
      "max_single_position": 0.095,
      "top_3_concentration": 0.235,
      "asset_class_max": 0.287
    },
    "liquidity_metrics": {
      "portfolio_liquidity_score": 0.94,
      "illiquid_allocation": 0.04,
      "days_to_liquidate_95": 2.5
    },
    "compliance_checks": {
      "max_single_weight_limit": 0.15,
      "max_class_weight_limit": 0.50,
      "max_crypto_limit": 0.05,
      "leverage_limit": 1.0
    }
  },
  "predecessor_actions": {
    "all_experts_and_agents": "详见上方各智能体输出"
```

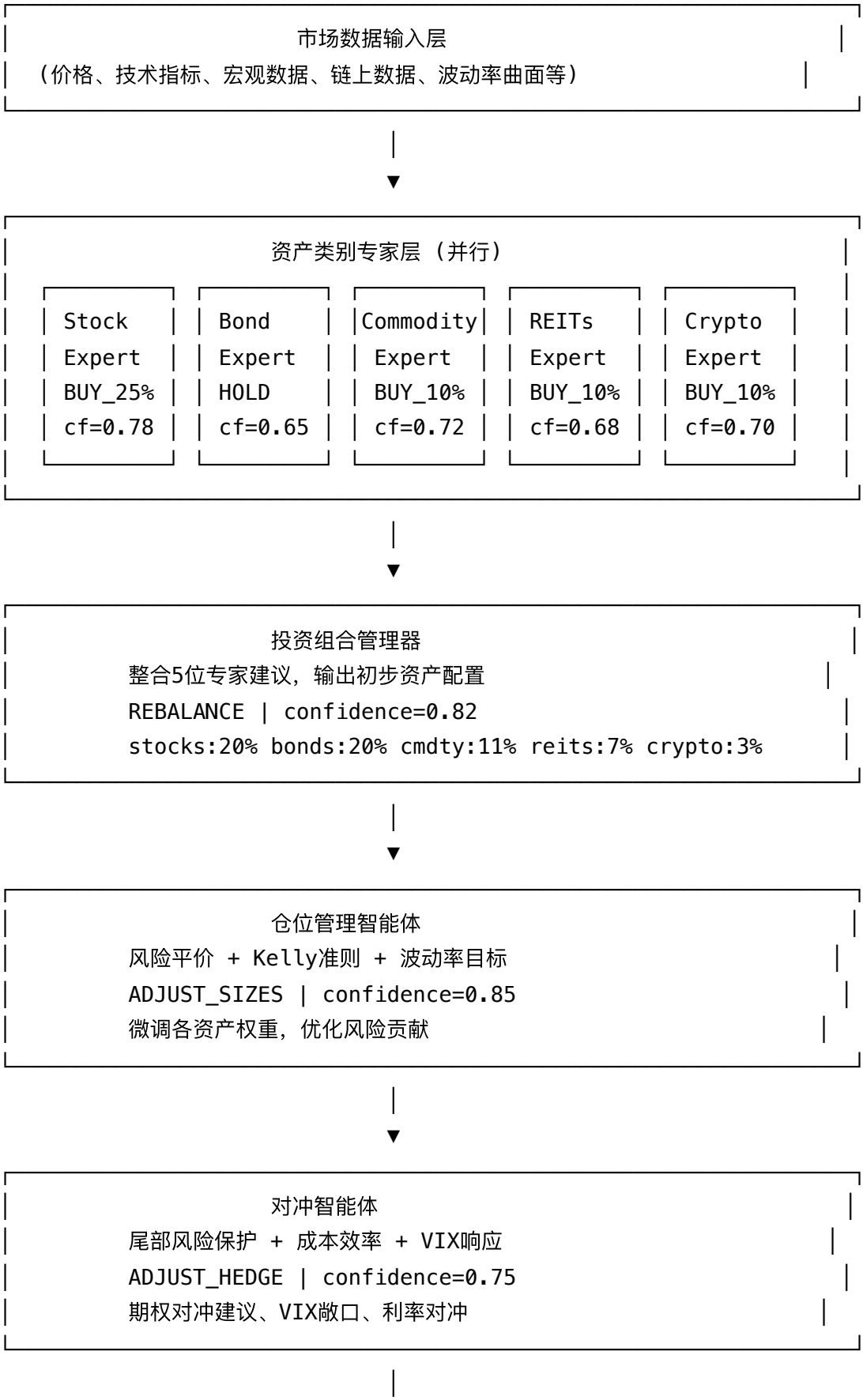
```
  },
  "risk_limits": {
    "max_portfolio_vol": 0.15,
    "max_var_95": -0.025,
    "max_drawdown_tolerance": -0.15,
    "min_sharpe_target": 1.0
  }
}
```

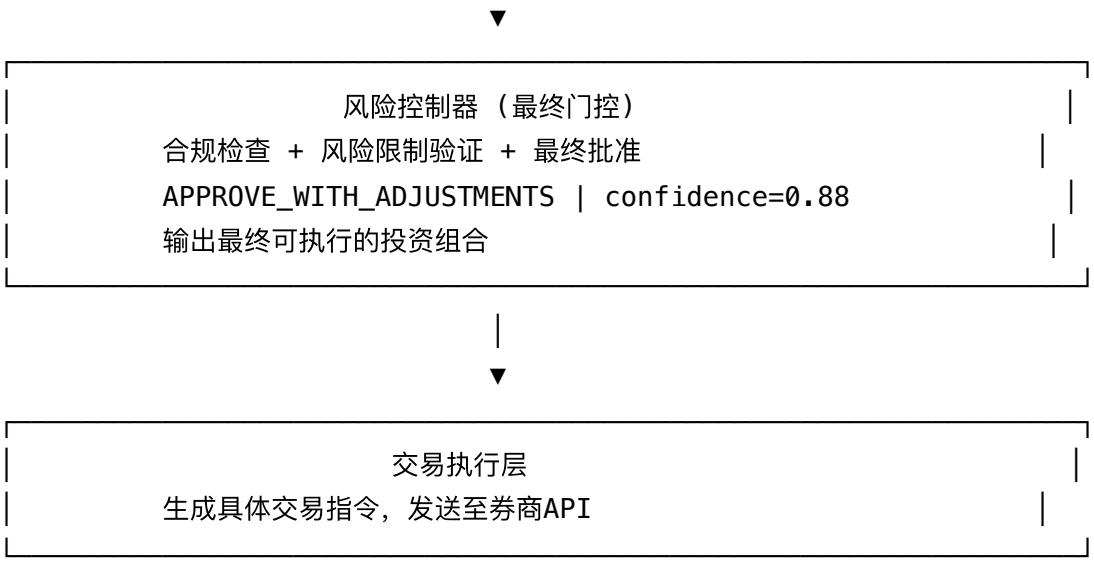
输出示例：

```
{
  "action": "APPROVE_WITH_ADJUSTMENTS",
  "action_id": 7,
  "confidence": 0.88,
  "final_approved_portfolio": {
    "SPY": 0.095,
    "QQQ": 0.052,
    "VTI": 0.04,
    "TLT": 0.048,
    "IEF": 0.09,
    "BND": 0.07,
    "GLD": 0.075,
    "USO": 0.02,
    "VNQ": 0.05,
    "BTC": 0.018,
    "ETH": 0.008,
    "CASH": 0.354
  },
  "adjustments_made": [
    {
      "asset": "BTC",
      "original": 0.02,
      "adjusted": 0.018,
      "reason": "略降加密货币敞口以保持风险预算"
    },
    {
      "asset": "ETH",
      "original": 0.01,
      "adjusted": 0.008,
      "reason": "与BTC同步调整"
    }
  ],
  "risk_validation": {
    "vol_check": {"limit": 0.15, "actual": 0.116, "status": "PASS"},
    "var_check": {"limit": -0.025, "actual": -0.017, "status": "PASS"},
    "concentration_check": {"limit": 0.15, "actual": 0.095, "status": "PASS"},
    "liquidity_check": {"min_score": 0.85, "actual": 0.94, "status": "PASS"},
    "crypto_limit_check": {"limit": 0.05, "actual": 0.026, "status": "PASS"}
  },
  "risk_decomposition": {
    "systematic_risk": 0.65,
    "idiosyncratic_risk": 0.35,
    "factor_exposures": {
```

```
      "market": 0.55,
      "size": 0.12,
      "value": -0.08,
      "momentum": 0.22,
      "volatility": -0.15
    }
  },
  "reasoning": "组合整体风险指标均在限制范围内：波动率11.6%<15%，VaR -1.7%<-2.5%，最大单一持仓9.
  "monitoring_alerts": [
    {
      "metric": "equity_concentration",
      "current": 0.187,
      "threshold": 0.25,
      "status": "WATCH",
      "note": "股票集中度适中，持续监控"
    }
  ],
  "key_factors": [
    {"factor": "risk_limit_compliance", "score": 0.92, "weight": 0.30},
    {"factor": "diversification_quality", "score": 0.85, "weight": 0.25},
    {"factor": "liquidity_adequacy", "score": 0.94, "weight": 0.25},
    {"factor": "tail_risk_mitigation", "score": 0.78, "weight": 0.20}
  ]
}
```

12. 智能体决策流程图


市场数据输入层
（价格、技术指标、宏观数据、链上数据、波动率曲面等）
资产类别专家层（并行）
Stock Expert
BUY_25%
cf=0.78
Bond Expert
HOLD
cf=0.65
Commodity Expert
BUY_10%
cf=0.72
REITs Expert
BUY_10%
cf=0.68
Crypto Expert
BUY_10%
cf=0.70
投资组合管理器
整合5位专家建议，输出初步资产配置
REBALANCE | confidence=0.82
stocks:20% bonds:20% cmdty:11% reits:7% crypto:3%
仓位管理智能体
风险平价 + Kelly准则 + 波动率目标
ADJUST_SIZES | confidence=0.85
微调各资产权重，优化风险贡献
对冲智能体
尾部风险保护 + 成本效率 + VIX响应
ADJUST_HEDGE | confidence=0.75
期权对冲建议、VIX敞口、利率对冲



附录A: 训练硬件要求

配置项	推荐值
GPU	A100-SXM4-80GB 或更高
GPU显存	≥80GB
系统内存	≥128GB
存储	≥500GB NVMe SSD
预计训练时间	~6-12小时 (180步)

附录B: 关键文件路径

```
FinSage/
├── finsage/rl/
│   ├── config.py                  # 配置定义
│   ├── reward_functions.py        # 奖励函数
│   ├── shared_expert_manager.py   # 智能体管理
│   ├── critic.py                  # Critic网络
│   ├── data_bridge.py             # 数据桥接
│   └── marft_integration.py       # MARFT集成
├── scripts/
│   └── train_with_real_data_v4.py # V4训练脚本
└── docs/
    └── MARFT_V4_Framework_Report.md # 本文档
```

文档版本: 1.0

最后更新: 2024-12-16