Chapter-by-Chapter Explanation of the Paper

Title: *Nonlinear Information Fusion Algorithms for Data-Efficient Multi-Fidelity Modelling*
Authors: P. Perdikaris, M. Raissi, A. Damianou, N. D. Lawrence, G. E. Karniadakis

1. Introduction

Key Problem: Traditional multi-fidelity models (e.g., Kennedy & O'Hagan's linear autoregressive scheme) struggle to capture **nonlinear** or **space-dependent cross-correlations** between low- and high-fidelity data. Low-fidelity models may provide erroneous trends outside their validity regime, limiting their utility.
Solution: Introduce a **nonlinear autoregressive Gaussian process (NARGP)** framework that generalizes linear autoregressive models by learning complex cross-correlations through deep GP-inspired structures.

Core Idea: Replace the linear scaling factor ρ in Kennedy & O'Hagan's model with a **nonlinear mapping** $z_{t-1}(f_{t-1}(x))$, modeled as a Gaussian process. This allows the algorithm to capture nonlinear dependencies while maintaining computational tractability.

Supplementary Material:

- **Gaussian Process Regression**: Basics in Rasmussen & Williams (2006).
- **Kennedy & O'Hagan (2000)**: Original linear autoregressive multi-fidelity model DOI:10.1093/biomet/87.1.1.

2. Methods

Model Structure:

1. **Recursive GP Framework**:
   - For fidelity levels $t = 1, \ldots, s$, model outputs as:

$$f_t(x) = g_t\left(x, f_{t-1}(x)\right) + \delta_t(x),$$

   where g_t is a GP mapping inputs x and lower-fidelity outputs $f_{t-1}(x)$ to higher-fidelity outputs.

- **Kernel Design**: Use a composite kernel to decompose contributions from input space and lower-fidelity outputs:

$$k_{t_g} = k_{t_\rho}(x, x') \cdot k_{t_f}(f_{t-1}(x), f_{t-1}(x')) + k_{t_\delta}(x, x'),$$

  where k_{t_ρ}, k_{t_f}, k_{t_δ} are squared-exponential kernels with ARD weights.

2. **Training Workflow**:
   - **Step 1**: Train a standard GP on the lowest-fidelity data.
   - **Step 2**: Sequentially train GPs at higher fidelity levels using the posterior mean of the previous level as an input.
   - **Step 3**: Propagate uncertainty through Monte Carlo integration for predictions.

Key Innovation: Avoids the intractability of deep GPs by using deterministic posterior means from previous levels, simplifying training to standard GP regression.

Supplementary Material:

- **Automatic Relevance Determination (ARD)**: Explained in Bishop's *Pattern Recognition and Machine Learning*.
- **Monte Carlo Uncertainty Propagation**: Tutorial in Girard et al. (2003).

3. Results

Benchmark Problems:

1. **Pedagogical 1D Example**:
   - **Low-fidelity**: $f_l(x) = \sin(8\pi x)$.
   - **High-fidelity**: $f_h(x) = (x - \sqrt{2})f_l^2(x)$.
   - **Result**: NARGP captures nonlinear cross-correlations (Figure 2a), outperforming AR1 (Figure 2b).
2. **Branin Function (3-Level)**:
   - **Low/medium/high-fidelity**: Complex transformations of the Branin function.
   - **Result**: NARGP achieves $\mathbb{L}_2$ error = 0.023 vs. AR1's 0.112 (Figure 7).
3. **Mixed Convection Flow**:
   - Combines experimental correlations (low-fidelity) and Navier-Stokes simulations (high-fidelity).
   - **Result**: NARGP reduces prediction error by leveraging nonlinear correlations, even in opposing flow regimes (Figure 10).

Key Insight: NARGP outperforms linear autoregressive models in accuracy, especially in regions where low-fidelity data are sparse or misleading.

Supplementary Material:

- **Branin Function**: Benchmark details in Forrester et al. (2008).
- **Latin Hypercube Sampling**: Explained in McKay et al. (1979).

4. Discussion & Conclusion

Advantages of NARGP:

1. **Flexibility**: Captures nonlinear/space-dependent correlations without deep GP complexity.
2. **Efficiency**: Maintains $\mathcal{O}(n^3)$ training cost (same as AR1).
3. **Robustness**: Safeguards against misleading low-fidelity trends.

Limitations:

- Assumes **noiseless data**; extensions to noisy data require semi-supervised GPs.
- Nested experimental designs ($D_t \subseteq D_{t-1}$) may restrict flexibility.

Future Work:

- Integration with scalable GP approximations (e.g., sparse GPs).
- Applications to high-dimensional/real-time systems.

Supplementary Material:

- **Sparse GPs**: Snelson & Ghahramani (2005).
- **Deep GPs**: Damianou & Lawrence (2013).

Study Recommendations

1. **Prerequisite Knowledge**:
    - Gaussian processes, kernel methods, Bayesian inference.
    - Basics of multi-fidelity modeling (Kennedy & O'Hagan's work).
2. **Hands-On Practice**:
    - Implement NARGP using GPy (code available here).
    - Experiment with the Branin function or pedagogical 1D example.

3. **Further Reading**:
    - Peherstorfer et al. (2016) (survey of multi-fidelity methods).
    - Deep Gaussian Processes (advanced topic).

Let me know if you need help dissecting specific equations or figures!