Neural network fitting with categorical variables as conditioners: literature review (2025)

Introduction

In many financial or industrial applications, models must predict an outcome from numerical features while controlling for a categorical variable such as sector, exchange, or experimental condition. In the **fitting engine** described by the user, the categorical variable is **not used as a direct feature** but instead *conditions* the model so that the effects of other features differ across categories. Rather than one-hot-encoding the category or adding dummy variables (which can lead to high-dimensional sparse inputs), modern neural network architectures can **modulate feature processing** based on a conditioning signal. This review summarises literature up to 2025 on methods for category-conditioned neural networks, covering challenges of high-cardinality categorical features, mixture-of-experts and gating architectures, conditional computation frameworks, feature-wise modulation techniques, contextual feature selection, random-effects integration and meta-learning approaches.

Challenges of high-cardinality categorical variables

High-cardinality categories often lead to sparse inputs and overfitting when naïvely one-hot encoded. A 2023 survey on deep learning for tabular data notes that standard encodings can impose artificial ordering or produce high-dimensional sparse vectors, making it difficult for neural networks to generalisearxiv.org. In many datasets there are also repeated measures (e.g. multiple observations for each individual) or correlated clusters. The LMMNN model for integrating random effects treats the high-cardinality category as a random variable within a mixed-effects framework, allowing the network to model correlated data without overfittingjmlr.org. Modelling categories as *random effects* rather than fixed inputs can thus improve prediction and reduce parameter explosion.

Mixture-of-Experts (MoE) and gating architectures

Mixture-of-Experts is a classic strategy for conditional computation: a **gating network** assigns each input to one or more expert networks, whose outputs are combined. Several variants are relevant for conditioning on a categorical variable.

- **Classical MoE** – A survey on MoE emphasises the role of the gating function, which typically uses a softmax to produce a probability distribution over expertsarxiv.org. The gating network can be linear or non-linear (e.g. using cosine routers), and its design strongly influences performancearxiv.org. When the conditioning variable is categorical, the gating network can map category-specific features to mixture weights; categories with similar patterns can share experts.
- **Gumbel-Softmax gating for tabular data** – A recent (2025) MoE model for tabular data introduces a **Gumbel-Softmax gating function** that regularises the gating distribution and prevents expert collapse. The

gating network outputs weights for each expert; the temperature parameter controls the degree of randomness and can be annealed during trainingarxiv.org. This architecture uses piecewise-linear embeddings for numerical features and one-hot encoding for categories, showing that MoE can handle mixed data types while conditioning experts on categoriesarxiv.org.

- **Multi-gate mixture-of-experts (MMoE)** – For multitask learning and recommendation, MMoE employs separate gating networks per task. An article by Kuaishou notes that gating networks determine how inputs are dispatched to experts and that poor design can lead to *expert collapse*arxiv.org. This concept translates to category-conditioning: each category can have its own gate so that different experts process the same features differently.
- **Conditional channel-gated networks** – The **Conditional Channel Gated Network** for continual learning attaches lightweight gating modules to each convolutional layer. Each gating module takes the layer's feature map as input and outputs a binary vector (via a Gumbel-Softmax approximation) indicating which channels to useopenaccess.thecvf.com. New gating modules are instantiated for each new task, but each module is small (an MLP with 16 units)openaccess.thecvf.com. A sparsity loss encourages each gate to select a minimal set of channelsopenaccess.thecvf.com, enabling dynamic capacity depending on the category or task.
- **User-Resizable Residual Networks (URNet)** – URNet places a **Conditional Gating Module (CGM)** before each residual block. The CGM takes two inputs: the previous layer's features and a scale parameter; it outputs a binary gate that decides whether to execute the blockarxiv.org. At test time the CGM uses a hard step function so blocks can be selectively skipped, adjusting computational cost without large accuracy lossarxiv.org. While URNet focuses on resource scaling rather than category-conditioning, the same gating mechanism could be driven by a categorical signal to modulate network depth.
- **Gated Linear Networks (GLNs)** – GLNs are backpropagation-free architectures that use **contextual gating** to select weight vectors based on side information. Each neuron receives side information z (which may be categorical) and passes it through context functions c(z) that select one of several weight vectorscdn.aaai.org. Thus, neurons specialise their weighting of inputs based on the category. The gating is performed using half-space classifiers, and the authors show GLNs have universal approximation capabilities for sufficiently many contextscdn.aaai.org.
- **Channel or layer gating for dynamic inference** – Other architectures (e.g. URNet or ConvNet-AIG) use conditional gating modules to decide

whether to apply certain layers or channels based on input or context. These gating modules can be conditioned on a category variable to enable per-category inference graphs.

Conditional computation frameworks and dynamic routing

Conditional computation generalises MoE by allowing a network to **sparsely activate** certain neurons, channels or layers based on a conditioning input. A general formalism by Goyal et al. defines different forms of dynamic sparsity (input, width and depth). The gating decision depends on a **conditioning value** u, which can be equal to the main input or an auxiliary variablearxiv.org. In the dynamic-width case, a subsampling operation selects some experts given $u$arxiv.org. For category-conditioning, the categorical variable acts as u, controlling which experts or neurons fire.

Conditional convolutional networks dynamically activate convolution kernels based on intermediate representations. A 2021 conditional CNN learns to route each sample through different kernels, effectively partitioning the network by hidden modalities without prior knowledgeopenaccess.thecvf.com. Such dynamic routing can be adapted to categories, allowing different categories to use different filters.

Feature-wise modulation and conditional normalisation

Many conditioning methods alter **feature activations** rather than selecting experts. These methods view the conditioning variable as generating modulation parameters (scales or shifts) applied to features.

- **Feature-wise Linear Modulation (FiLM)** – FiLM uses a **FiLM generator** that outputs per-feature scale γ and shift β parameters from conditioning input x. These parameters modulate intermediate activations via an affine transformation $\mathrm{FiLM}(F_{i,c}|\gamma_{i,c},\beta_{i,c}) = \gamma_{i,c} F_{i,c} + \beta_{i,c}$ cdn.aaai.org. Modulation can be conditioned on the same input or on an auxiliary variable; the generator is typically a small neural network. FiLM layers have low computational cost and provide fine-grained control over the networkcdn.aaai.org. Originally developed for visual reasoning, FiLM generalises **conditional normalisation**, which replaces batch-norm parameters with functions of the conditioning inputcdn.aaai.org.
- **Conditional Batch/Instance/Layer Norm** – Conditional normalisation methods replace the scale and shift parameters of batch or layer normalisation with functions of the conditioning variable. In Modulating early visual processing by language, a **Conditional Batch Normalisation (CBN)** layer uses language embeddings to compute the scale and bias of batch normalization, enabling the network to incorporate linguistic signals in early convolutional layersarxiv.org. Similarly, Dynamic Layer Norm and Conditional Instance Norm adapt normalisation parameters for style transfer, speech recognition and cross-modal tasks.
- **Conditional networks** – *Conditional networks* integrate an auxiliary

network that processes metadata (e.g. geographic coordinates) and generates scale & shift parameters for the main network's layersopenreview.net. The auxiliary network's features are passed through learned functions to produce conditional normalisation parameters, enabling out-of-distribution generalisationopenreview.net. The architecture is flexible and can learn to ignore irrelevant context.

- **Hypernetworks** – Hypernetworks generate the weights of a target network based on a conditioning input. A review notes that hypernetworks provide soft weight sharing across tasks, dynamic architectures and parameter efficiency; they can be conditioned on data or tasksarxiv.org. When the categorical variable serves as the condition, a hypernetwork can produce different weights for each category or a continuous interpolation between categoriesarxiv.org.
- **Contextual Adaptation via Meta-Learning (CAVIA)** – CAVIA separates network parameters into **context parameters** (low-dimensional vectors that are adapted for each task) and shared parameters. The context parameters are concatenated to network layers and modulate computation. During meta-training, the context parameters are updated for each task; this reduces overfitting and makes adaptation interpretablearxiv.org. In category-conditioned networks, each category would have its own context vector.

Contextual feature selection and adaptive interactions

Another line of work performs **feature selection conditioned on context** so that different features are used for different categories.

- **Contextual Feature Selection with Conditional Stochastic Gates (c-STG)** – c-STG introduces **conditional Bernoulli gates** for each feature. A hypernetwork maps the contextual variables (e.g. category) to the logits of the Bernoulli gates, representing the probability that a feature is selectedarxiv.org. To train the model with gradient descent, the Bernoulli distribution is relaxed to a continuous distribution using a Gaussian approximationarxiv.org. This mechanism learns context-dependent sparsity patterns, selecting features only when useful for a given category.
- **Gated Adaptive feature Interaction Network (GAIN)** – In GAIN, each feature field has a gate that decides whether to participate in interactions for click-through rate prediction. Gates are continuous (via Gumbel-Softmax) and updated via gradient descent; the network adaptively selects features that matter for the current contextpmc.ncbi.nlm.nih.gov. This approach can be adapted to category conditioning by associating gates with categories.
- **TabNet** – TabNet uses sequential attention to select a subset of features at each decision step. The attention masks indicate which features are

used at each step, enabling per-instance feature selection. The authors emphasise that instance-wise feature selection leads to efficient use of capacity and interpretabilityarxiv.org. While TabNet does not explicitly incorporate category conditioning, the attention mechanism can implicitly focus on different features for different categories.

Random-effects neural networks and mixed models

When the category variable represents clusters of correlated samples (e.g. repeated measurements per subject or instrument), it is natural to treat the category as a **random effect**. Simchoni & Rosset propose integrating random effects into deep neural networks (LMMNN). They treat the category variable as a random effect with unknown variance components and use the **Gaussian negative log-likelihood (NLL)** as the loss functionjmlr.org. This approach handles high-cardinality categorical features without introducing a large number of fixed parameters and can be combined with any neural architecture.

Other works treat high-cardinality categorical features as random effects in a regression setting and show that this yields better predictive performance than one-hot encodingproceedings.neurips.cc. This perspective is important for quant researchers: if the categorical variable corresponds to a large number of instruments or sectors, modelling it as a random effect can regularise the network and avoid overfitting.

Discussion and design guidelines

The reviewed literature offers several strategies for integrating a categorical variable as a **conditioner** rather than as a simple input:

1. **Choose a conditioning mechanism** based on model goals:
    - If the goal is to route samples to specialised sub-networks, **MoE** or **gating** architectures are appropriate. They allow different categories to use different experts or channels but require careful gate design (e.g. Gumbel-Softmax or regularised gating to avoid expert collapsearxiv.org).
    - For per-feature modulation, **FiLM** or **conditional normalisation** layers are effective. They require only a small number of additional parameters per channel and can be easily added to existing architecturescdn.aaai.org.
    - If the category is high-cardinality with limited observations per category, modelling it as a **random effect** with a mixed-effects loss is desirablejmlr.org.
    - When a small context vector is sufficient, **CAVIA** or **hypernetwork** approaches can generate category-specific weights or context parametersarxiv.org.
2. **Gating networks should be robust**. To prevent gating collapse, use temperature annealing or regularisation (e.g. entropic regularisers or Gumbel-Softmax). Multi-gate architectures may assign separate gates

per category or feature group to provide more flexibilityarxiv.org.

3. **Feature selection can be context-dependent**. Methods like c-STG or TabNet select features or interactions based on the categorical context, which can lead to interpretable models and improved efficiencyarxiv.org. However, careful relaxation of the Bernoulli gates is needed for differentiabilityarxiv.org.
4. **Normalization and modulation**. Conditional batch norm or FiLM layers can incorporate the category information at multiple depths of the network, allowing coarse-to-fine adaptation. These layers add negligible overhead and have been shown to improve multimodal reasoning and out-of-distribution generalizationopenreview.net.
5. **Use hypernetworks for full weight adaptation**. When category differences are large (e.g. different instruments with distinct dynamics), hypernetworks can generate entirely different weights for each category while sharing underlying structurearxiv.org.
6. **Consider computational cost**. Gating and conditional computation allow dynamic adjustment of computational resources. URNet demonstrates that conditional gating can reduce computation while maintaining accuracyarxiv.org, which may be valuable when deploying models across many categories with varying resource budgets.

Conclusion

Incorporating categorical variables as *conditioners* in neural networks leads to architectures that adapt their behaviour based on the category rather than treating the category as a static input. Recent research across mixture-of-experts, gating networks, conditional computation, feature-wise modulation, contextual feature selection, hypernetworks and mixed-effects modelling provides a rich toolkit for quant researchers. Selecting an appropriate conditioning mechanism depends on the problem context: whether categories correspond to tasks, clusters with few samples, or major shifts in feature relevance. By leveraging these methods, neural network fitting engines can achieve more accurate and interpretable models in settings with complex categorical heterogeneity.