Using Categorical Variables as Conditioners in Machine Learning (2016-2025)

Introduction

Machine-learning models often treat categorical variables as discrete labels and encode them as high-dimensional one-hot vectors or as arbitrary integer IDs. This simple representation discards proximity relationships between categories and leads to sparsity and high memory usage. When tabular datasets contain many categories and limited examples per category, one-hot encoding can also hurt generalization because each category appears in only one input pattern. To mitigate these issues, researchers have explored **conditioning techniques** that transform categorical values into more informative continuous representations and use them to modulate neural networks or improve other feature representations. This literature review examines the evolution of such methods, focusing on *entity embeddings*—an approach that learns dense vector representations for categories—and surveys tree-based encodings, feature-interaction models, attention-based transformers, gating mechanisms, hypernetworks, and mixed-effect models developed between 2016 and 2025.

Entity Embeddings of Categorical Variables

The landmark paper *Entity Embeddings of Categorical Variables* (2016) introduced **entity embeddings**, a method that learns a dense vector in Euclidean space for each category using supervised neural network trainingarxiv.org. Instead of one-hot vectors, each categorical value is mapped via a learned lookup table to a low-dimensional embedding. These embeddings are optimized jointly with the downstream task and capture statistical relationships among categories. Similar categories (e.g., nearby postal codes or products with similar sales patterns) acquire similar embedding vectors, enabling generalization across rare categories and revealing intrinsic structuresarxiv.org. The authors demonstrated that replacing one-hot encodings with entity embeddings reduces memory usage, accelerates training, and improves prediction performance. Embeddings can also be visualized to cluster categories and define distance measuresarxiv.org.
The core idea is depicted in the conceptual diagram below, where categories are represented by colored shapes and mapped into points in a continuous space. This continuous representation provides a smooth “metric” over categories and can be used as an input to any downstream model.

Training entity embeddings

To learn entity embeddings, a neural network is built where the categorical input passes through an embedding layer (a weight matrix) and the embeddings are concatenated with numerical features. The network is trained end-to-end on the target task, allowing the embeddings to reflect the task-specific similarity of categories. The authors compared this method with one-hot encoding on multiple datasets and showed improved accuracy and generalizationarxiv.org. Importantly, entity embeddings can be reused: once trained on a large dataset, the

embeddings provide a prior that can boost performance on related tasks with limited data.

Impact and subsequent developments

Entity embeddings spurred numerous follow-up works in tabular deep learning. They inspired research on *contextual embeddings* and *transformer-based models* that not only embed categories but also capture interactions among them. They also motivated tree-based methods to derive continuous representations based on target statistics and other statistical measures. The next sections discuss these developments.

Tree-Based Encodings and Ordered Target Statistics

Target encoding and leakage

A common alternative to one-hot encoding is **target encoding**, where each category is replaced by a statistic such as the mean target value for that category. While effective, this method can introduce *target leakage* if the statistic uses the same target being predicted. The CatBoost algorithm addresses this issue using **ordered target statistics**. For each category, CatBoost iterates over a random permutation of the training data and computes the target statistic using only preceding examples. This ensures that no future (unseen) target information leaks into the encodingarxiv.org. The paper shows that this *ordered TS* uses all data while preserving unbiasedness and yields significant improvementsarxiv.org.

CatBoost feature combinations

CatBoost also constructs **combinations of categorical features** to capture higher-order dependencies. At each decision tree split, the algorithm concatenates categorical features used in previous splits with the remaining features, forming interaction features. These combinations are then converted into ordered target statistics on the flyarxiv.org. By greedily generating combinations during tree growth, CatBoost efficiently explores interactions without enumerating all possible combinations. This technique improves performance on datasets with complex categorical relationships.

GBDT-based feature gating and early selection

Recent works extend tree-based ideas to gating mechanisms. T-MLP trains a gradient boosting decision tree (GBDT) and uses the frequency of feature access for each sample to construct a **sample-specific binary mask**. This mask gates the embedding layer of a multilayer perceptron (MLP), allowing the model to focus on important categorical features. Experiments show that the GBDT gate reduces the number of activated embedding parameters without sacrificing accuracyarxiv.org. Another method, Adaptive Early Feature Selection (AEFS), trains an auxiliary model to select a subset of raw features before embedding. The main model then uses only the selected features, reducing the embedding layer size by 37.5 % while matching the accuracy of late selection methodsarxiv.org.

Cross Feature Interaction Models

Explicit cross layers and mixture-of-experts

Learning interactions between features is crucial when categories interact in

non-additive ways. Deep & Cross Networks (DCN) introduced *cross layers* that compute polynomial feature interactions by repeatedly applying the operation
x
l
+
1
=
x
0

x
l
⊤
w
+
b
+
x
l
x_{l+1}=x_0\,x_l^\top w + b + x_l
xl+1 =x0 xl⊤ w+b+xl . The second version (DCN-V2) replaces the single cross layer with a **mixture of low-rank experts**. Several low-rank cross modules are trained in parallel, each capturing different interaction patterns. A **gating function** (sigmoid or softmax) produces input-dependent weights that combine the expert outputs, making the model more expressive yet efficientarxiv.org.

Factorization Machines and xDeepFM

Factorization Machines (FMs) and their extensions (e.g., Field-aware FMs) model pairwise interactions using latent vectors for each feature. xDeepFM combines an FM with a neural network to capture both low- and high-order interactions by stacking cross networks and deep networks. These models treat categorical variables using embeddings and can be viewed as early precursors to cross-feature conditioning.

Gated interaction networks

The **Gated Adaptive Feature Interaction Network (GAIN)**, used in click-through rate prediction, comprises a cross module with **multiple gated units**. Each unit computes high-order interactions and uses gating to drop meaningless interactions while preserving informative onespmc.ncbi.nlm.nih.gov. The introduction notes that manually constructing cross features is challenging and that deep neural networks implicitly learn interactions but may not capture complex high-order terms. GAIN shows that gating can adaptively select meaningful interactions with minimal overheadpmc.ncbi.nlm.nih.gov.

Another recent model, **GANDALF**, introduces a **Gated Feature Learning Unit (GFLU)** that uses update and reset gates to modulate how much of the candidate feature representation is incorporated and how much previous information to

forget. Feature masks are applied only to gates to avoid harming gradient flow. Stacking GFLUs yields a hierarchical feature selection processarxiv.org.

Contextual Embeddings and Transformer-Based Models

TabTransformer

While entity embeddings treat each categorical feature independently, the **TabTransformer** uses **self-attention** to create contextual embeddings. The model first maps each categorical value to a vector (like entity embedding) and then applies Transformer layers across categorical tokens. This allows the model to learn contextualized representations where each embedding considers correlations with other categorical features. The authors show that MLP embeddings are context-free and limited; the Transformer produces **contextual embeddings** that improve accuracy and robustness to missing/noisy dataarxiv.orgarxiv.org. When only numerical features are present, TabTransformer reduces to a multi-layer perceptron, demonstrating its flexibilityarxiv.org.

LLM embeddings for tabular data

A 2025 study proposes converting tabular data into sentences and encoding them with **large language models (LLMs)**. The paper notes that traditional tabular methods encode numerical and categorical features separately; the new approach represents each feature and its value as tokens and passes the resulting sentence through an LLM. This allows the model to capture interactions between features and values and to reuse embeddings across tasksarxiv.org. By leveraging pre-trained LLMs, the method enhances performance and generalizes across tables.

TabNet, TabTransformer derivatives and sparse attention

TabNet uses sequential attention to select a subset of features at each decision step, learning a sparse mask that determines which features to focus on. TabNet inspired models like **TabNSA (Native Sparse Attention)**, which uses hierarchical sparse attention to select relevant subsets of features dynamicallyarxiv.org. These models treat categorical features via embeddings and employ attention or gating to emphasize important interactions and ignore irrelevant ones.

Hypernetworks and Conditioning

Feature-wise linear modulation (FiLM)

FiLM layers perform a feature-wise affine transformation on neural activations conditioned on auxiliary inputs. Given an intermediate feature
F
i
,
c
F_{i,c}
Fi,c and conditioning input, FiLM computes scaling (
γ
\gamma
γ) and bias (

β
\beta
β) parameters and modulates the feature as
F
i
L
M
(
F
i
,
c
∣
γ
i
,
c
,
β
i
,
c
)
=
γ
i
,
c
F
i
,
c
+
β
i
,
c
\mathrm{FiLM}(F_{i,c}\mid \gamma_{i,c},\beta_{i,c})=\gamma_{i,c}F_{i,c}+
\beta_{i,c}
FiLM(Fi,c ∣γi,c ,βi,c)=γi,c Fi,c +βi,c cdn.aaai.org. The conditioning network
producing
γ
\gamma
γ and

β
\beta
β can be any architecture, enabling tasks like visual question answering and domain adaptationcdn.aaai.org. In tabular contexts, FiLM can modulate the processing of numerical features based on categorical variables.

HyperFusion and tabular-image fusion

The **HyperFusion** framework employs a **hypernetwork** to integrate electronic health record (EHR) variables with imaging data. The hypernetwork takes tabular features (including categorical variables) as input and generates weights for the primary image-processing network. Thus, the imaging network's parameters are conditioned on tabular informationarxiv.org. This approach illustrates how high-dimensional features can be conditioned on categorical context to improve multimodal tasks.

Mixed-effect neural networks

High-cardinality categorical variables may induce correlations and hierarchical structures. *Latent Mixed Models Neural Networks (LMMNN)* integrate random effects into deep networks by combining mixed-effect models with neural components. The framework treats high-cardinality categories as random effects, learning fixed and random effect parameters via maximum likelihood. It acknowledges that standard DNNs assume independent observations, whereas categorical variables introduce correlations; by modeling these random effects explicitly, LMMNN improves predictive performancejmlr.org.

Feature Selection and Disentanglement

SwitchTab and disentangling mutual and salient features

SwitchTab uses an asymmetric encoder–decoder to decouple mutual (shared) and salient (sample-specific) features in tabular data. During training, each sample's features are encoded into a general embedding that is split into *salient* and *mutual* subspaces. The decoder reconstructs the original data by swapping the salient and mutual embeddings between samples, forcing the encoder to disentangle shared and unique informationarxiv.org. Although not directly conditioning categorical variables, SwitchTab demonstrates how disentangling features can improve downstream models and could be applied to condition categorical features separately from numerical ones.

Early gating and mask learning

Models like T-MLP and AEFS (discussed above) apply gating or feature selection early in the pipeline, reducing the cost of embeddings. These methods highlight a trend toward **adaptive feature masking**, where the model learns to select relevant categorical variables for each sample. The goal is to mitigate the computational burden of embedding high-dimensional categories and to prevent noise from irrelevant features.

Trends and Future Directions

1. **Contextual and self-supervised embeddings:** Inspired by TabTransformer, future models are exploring contextual embeddings

using self-attention and transformer architectures. Pre-training on unlabeled tabular data and fine-tuning for downstream tasks could leverage large unannotated datasets.

2. **LLMs for tabular data:** Converting tabular data to natural language and using large language models can capture complex feature interactions and support cross-table transferarxiv.org. Research is ongoing to balance the cost of large models with the benefits of improved representation.
3. **Hypernetworks and modulation:** FiLM and hypernetwork approaches show that conditioning neural networks on categorical variables can dynamically adjust parameters. Future work may explore more efficient hypernetworks for tabular tasks and unify feature selection with parameter generation.
4. **Random-effect and hierarchical models:** Mixed-effect neural networks like LMMNN address correlated data induced by categorical variablesjmlr.org. Extending these ideas to hierarchical structures and domain adaptation is a promising direction.
5. **Gating and sparse attention:** Many recent models employ gating or sparse attention to select relevant features. Techniques like GFLU, TabNSA, and gating networks improve model interpretability and efficiency. Future work may combine gating with contextual embeddings to further refine feature conditioning.

Conclusion

The past decade has seen significant progress in leveraging categorical variables beyond one-hot encodings. Entity embeddings introduced the idea of learning dense vectors that capture similarity between categories, enabling better generalization and visualizationarxiv.org. Tree-based methods like CatBoost use ordered target statistics and feature combinations to reduce bias and capture interactionsarxiv.orgarxiv.org. Cross-feature models and gated networks explicitly model interactions and adaptively select meaningful combinationsarxiv.orgpmc.ncbi.nlm.nih.gov. Transformer-based models provide contextual embeddings and robust representationsarxiv.org, while hypernetwork and mixed-effect approaches condition network parameters on categorical contextarxiv.orgjmlr.org. These developments illustrate the rich landscape of conditioning techniques for categorical variables and point towards a future where tabular models integrate self-supervised learning, hypernetworks, and adaptive feature selection for improved performance and interpretability.

Most of the methods discussed fall into two broad patterns:

- **Direct feature representation:** Many models simply map each categorical value into a dense vector (an “entity embedding”) and treat

that vector as just another input feature. CatBoost's ordered target statistics, factorization-machine variants and standard deep embedding models do exactly this—categorical information feeds directly into the predictive function alongside numerical variables, without actively modifying how other features are processed.

- **Conditional modulation:** A smaller subset of models use categorical variables to alter the computation applied to other features. Examples include FiLM layers and hypernetworks like HyperFusion, which generate scaling and bias parameters or entire weight matrices conditioned on categorical inputs. Gated models (T-MLP, GFLU in GANDALF, GBDT gates in T-MLP, AEFS) also learn masks or gates that turn on/off or reweight specific numerical features based on categorical context. Transformer-based architectures such as TabTransformer lie somewhere in between: they compute contextual embeddings by attending across categorical variables, so the representation of one category depends on its neighbours, but they still feed these embeddings into standard downstream layers rather than explicitly modulating other feature streams.