**YOLOv9** - Introduces two main innovations: **PGI**(Programmable Gradient Information) & **GELAN**(Generalized Efficient Layer Aggregation Network)

A diagram of a process

AI-generated content may be incorrect.

**a)** **CSPNet**- Cross Stage Partial Network, foundation of  YOLOv4, YOLOv5, YOLOv6, and used in **GELAN** in YOLOv9. **Split** the input. Sends one part through a deep block, the other is a shortcut. Concatenates and optionally transforms afterward

**b)** **ELAN** - Used in YOLOv7. Still splits input but**sends branches to multiple stacked conv paths of different depths**. Promote **deep gradient paths** without increasing complexity.

**c)** **GELAN** - Generalized ELAN, **backbone architecture** introduced in **YOLOv9.**Builds on ELAN but adds **flexibility** and **multi-path aggregation**. Allows **any block** (not just Conv) to be inserted like CSP, ResNet..etc

**Architecture of GELAN:**

* Input is split into multiple paths (like ELAN).
* Some branches are processed **deeply** through different **any-block \*n** stacks.
* Some paths skip or go through transition layers.
* There is denserinterconnection between paths.
* Final concatenation aggregates all intermediate and final results.

**Advantages of GELAN Over other YOLO:**

1. ELAN in YOLOv7 used only repeated conv layers. GELAN generalizes this to allow Res, CSP, DarkNet, or even custom blocks. This gives flexibility across different devices (edge, mobile, cloud).
2. GELAN splits features into parallel branches of different depths (like Inception but deeper). These are then concatenated and optionally passed through a transition layer.
3. GELAN achieves higher accuracy with fewer parameters and FLOPs than ELAN.
4. GELAN maintains stable accuracy even when changing block depth, unlike earlier YOLO models where deeper meant harder to train or overfitting.

**PGI** is a smart way to give extra supervision during training by using a helper branch that generates accurate gradients. It ensures the main YOLOv9 model learns meaningful features at all depths without slowing down inference.

A diagram of a dog

AI-generated content may be incorrect.

**a) PAN** - Path Aggregation Network -Traditional top-down and bottom-up multi-scale fusion.Problem: Information Bottleneck.

**b) RevCol - Reversible Columns** - Uses **reversible layers** to maintain full data flow in both directions. Problem: Heavy Cost.

**c) Deep Supervision**- Adds extra **prediction heads** at intermediate levels. Each level tries to predict small, medium, or large objects. Problem: Broken Information.

**d) PGI:**

1. **Main Branch**: The actual network used for inference.(Solid blue path)
2. **Auxiliary Reversible Branch**: Gives clean gradient signals to the main branch using extra supervision.(Blue dashed paths)
3. **Multi-level Auxiliary Info**: Merges supervision signals across **all scales** and **all objects**. (Purple dashed box)
4. **Advantages**: Maintains full semantic info. Only used during training. No inference cost. Works for both deep and lightweight models.

**How PGI Works:**

* During training, input passes through the **YOLOv9**network GELAN.
* PGI **adds an extra branch** that reverses features or supervises them mid-way.
* It generates **reliable gradient signals** from this auxiliary path.
* These gradients flow back to help the main branch **learn better features**.
* PGI helps the model learnwithout modifying the inference pipeline.