

# CausalFormer: Zaman Serilerinde Zamansal Nedensel Keşif için Yorumlanabilir Bir Transformer

## Giriş ve Arka Plan

Zaman serisi verilerindeki **nedensel ilişkilerin (neden-sonuç bağlantılarının)** keşfi, çeşitli alanlarda kritik bir görev haline gelmiştir. Örneğin, şehir içi ulaşım verilerinden trafik sıkışıklığına yol açan etkenlerin saptanması, iklim verilerinden hava durumu öngörülerini iyileştirme, finans zaman serilerinden piyasa dinamiklerini anlama gibi pek çok uygulamada **zamansal nedensellik keşfi** önemli rol oynar <sup>1 2</sup>. Zamansal nedensellik keşfi (temporal causal discovery), birden fazla zaman serisi arasında hangi serinin diğerini **neden** olarak etkilediğini (ve ne kadar gecikmeyle etkilediğini) belirlemeye çalışır. Burada *nedensellik*, genel olarak “bir değişkenin (neden) diğerinde meydana gelen değişimin (sonuç) kısmen sorumlusu ve sonucu etkilemesi” şeklinde tanımlanır <sup>2</sup>. Zaman serilerinde nedensellik, **zamansal öncelik kısıtına** tabidir, yani **nedensel etkinin sebebi zaman olarak daima sonucundan önce gelmelidir** <sup>3</sup>. Bu nedenle, nedensel ilişkiler genellikle geçmiş dönem verilerinin gelecekteki değerler üzerindeki etkisi biçiminde ele alınır. Ayrıca bazı durumlarda **eşzamanlı nedensellik** (instantaneous causality) de görülebilir: Örnek olarak ölçüm aralığının çok seyrek olması nedeniyle, bir serideki etki ile diğerindeki tepki aynı zaman damgasında ortaya çıkmış gibi gözükabilir <sup>4</sup>. Bunun yanı sıra her bir serinin kendi geleceğini etkilemesi anlamında **öz-nedensellik (self-causation)** de zaman serilerinde mümkündür <sup>4</sup>.

Geçmişte zamansal nedensellik keşfi için iki temel yaklaşım geliştirilmiştir: **istatistiksel yöntemler** ve **derin öğrenme tabanlı yöntemler** <sup>5</sup>. İstatistiksel yöntemler arasında **Granger nedenselliği** testi en bilinenlerindendir. Granger yaklaşımı, bir zaman serisinin geçmiş değerlerinin, başka bir serinin geleceğini tahmin etmede istatistiksel olarak anlamlı bir iyileşme sağlıyorsa o serinin nedensel etkisi olduğunu varsayar <sup>6</sup>. Örneğin, klasik bir doğrusal model olan VAR (Vector Auto-Regression), her serinin bugünkü değerini kendi geçmişi **ve diğer serilerin geçmişlerinin lineer birleşimi** olarak ifade eder; bu modelde eğer serinin geçmiş verileri, diğer bir serinin geleceğini açıklamada anlamlı katkı yapıyorsa Granger nedenselliği vardır denir <sup>7</sup>. İstatistiksel yaklaşımlar ayrıca koşullu bağımsızlık testlerine dayalı yöntemler (PC algoritması ve zamansal genellemesi PCMCİ gibi) <sup>8</sup>, nedensel modeli gürültü-terimleriyle tanımlayıp gürültü bağımsızlığı arayan yöntemler (örn. TiMINo) <sup>9</sup> ve olası yönlü grafik yapıları arasından skor optimizasyonu ile seçim yapan yöntemler (örn. DYNOTEARS) <sup>10</sup> gibi alt kategorilere de ayrılır. Ancak bu yöntemler genellikle doğrusal varsayımlar, düşük boyutlu veriler veya belirli istatistiksel koşullarla sınırlıdır. Gerçek dünya verilerindeki karmaşık ve doğrusal olmayan ilişkileri keşfetmekte zorlanabilirler <sup>11</sup>.

**Derin öğrenme tabanlı yöntemler**, karmaşık ve doğrusal olmayan nedensel ilişkileri yakalayabilmek için derin sinir ağlarının esnekliğinden faydalanmayı hedefler. Bu yöntemlerin temel stratejisi, bir **tahmin modeli** eğiterek, modelin iç yapısını veya parametrelerini analiz edip nedensel ilişkileri çıkarsamaktır. Özellikle **Granger nedenselliğini** derin öğrenme bağlamında kullanmak için, bir ağın çıktısına her bir girdinin ne derece katkı yaptığını ölçmeye çalışırlar. Ancak derin modeller genellikle “kara-kutu” niteliğinde olduğundan, bir girdinin katkısını izole etmek ve ölçmek ciddi bir yorumlanabilirlik zorluğudur. Bu yüzden literatürdeki mevcut derin öğrenme yaklaşımları, genellikle modelin yalnızca belirli bileşenlerinin parametrelerini inceleyerek nedensellik çıkarsamaya çalışır.

Örneğin, dikkat mekanizmasına sahip konvolüsyonel sinir ağlarını kullanan **TCDF (Temporal Causal Discovery Framework)** adlı çalışma, eğitilen modelin **konvolüsyon çekirdeklerini ve dikkat (attention) ağırlıklarını** analiz ederek hangi serinin hangisini etkilediğini belirlemiştir <sup>12</sup> <sup>13</sup>. Benzer biçimde, tamamen bağlantılı (fully-connected) MLP ve LSTM ağlarına dayalı cMLP, cLSTM yöntemleri de, ağ yapılarındaki belirli aktif bağlantıları veya ağırlık parametre büyüklüklerini nedensellik göstergesi olarak yorumlamışlardır <sup>14</sup>. Bazı yaklaşımlar ise zaman serileri arasında bilgi yayılımını modelleyip bu yayılımı kontrol eden kapı vektörleri veya difüzyon katsayılarını nedensel skorlar olarak kullanmıştır.

Ne var ki, **sadece belirli parametrelere bakarak nedensellik belirlemek çoğu zaman yetersiz kalmaktadır**. Derin bir modelin dikkat ağırlıkları veya konvolüsyon filtreleri, nedensel ilişkilerin bir bölümünü yansıtabilir ancak modelin geri kalan katmanlarındaki hesaplamalar da sonuçta önemli rol oynar <sup>12</sup> <sup>15</sup>. Örneğin, sadece konvolüsyon katmanını veya dikkat mekanizmasını inceleyen bir yöntem, bu katman sonrası gelen tam bağlantılı katmanların veya aktivasyon fonksiyonlarının etkisini ihmal eder. Oysa, tam bağlantılı katmanlar çıktılarının **yeniden ölçeklenmesi ve birleşimi** yoluyla dikkat katmanında yakalanan ilişkileri **pekiştirebilir veya zayıflatır**; ayrıca ağırdaki **bias terimleri**, girdi verilerinde açıkça görülmeyen sabit etkileri modele katabilir <sup>15</sup>. Nauta ve arkadaşlarının TCDF yöntemi bu açıdan “doğrudan ancak **eksik bir haritalama**” yapmaktadır <sup>13</sup> <sup>16</sup>. Literatürde Transformers gibi modellerin yorumlanabilirliği üzerine yapılan çalışmalar da, tek başına dikkat ağırlığı görselleştirmenin yeterli olmadığını, modeldeki diğer bileşenlerin de hesaba katılması gerektiğini göstermektedir <sup>13</sup> <sup>15</sup>. **Sonuç olarak, derin öğrenme modellerinin tamamını kapsayan bir yorumlama stratejisine ihtiyaç vardır**; aksi takdirde model içindeki nedensel temsilin bir kısmı gözden kaçabilir <sup>15</sup>.

## Problem Tanımı

Ele aldığımız problemde,  $N$  adet zaman serisinin bulunduğu bir sistemin nedensel yapısını keşfetmek istiyoruz. Bu amaçla ilk adım, bir **tahmin (forecasting) modeli** eğiterek zaman serilerinin nedensel temsiliğini öğrenmektir <sup>17</sup>. Formülasyon olarak, her bir  $x_j$  zaman serisi için model  $\hat{x}_j(t)$  şeklinde bir tahmin üretir. Bu tahmin, zaman serilerinin geçmiş gözlemlerinden hesaplanır. Özellikle,  $x_j$  serisinin **şu anki  $t$  anındaki değerinin** tahmini  $\hat{x}_j(t)$ , diğer tüm serilerin önceki ve mevcut zamandaki gözlemlerinin yanı sıra  $x_j$  serisinin kendi önceki değerleri kullanılarak elde edilir <sup>17</sup>. Bunu matematiksel olarak aşağıdaki şekilde ifade edebiliriz:

$$(1) \quad \hat{x}_j(t) = f\left(\bigcup_{i \neq j} x_i(\tau) \mid \tau = t_0, \dots, t; x_j(\tau) \mid \tau = t_0, \dots, t-1\right)$$

Burada  $t_0$  mevcut gözlem penceresinin başlangıcını belirtir. Yani  $\hat{x}_j(t)$  tahmini hesaplanırken, hedef serinin ( $j$ )  $t$  anına kadarki geçmişi ile diğer tüm serilerin  $t$  anına kadarki gözlemleri modele girdi olarak verilir <sup>17</sup>. Bu sayede model, her serinin diğerlerini öngörmeye ne ölçüde bilgi sağladığını öğrenmeye çalışır.

İkinci adımda, eğitilen tahmin modelinden yararlanarak zaman serileri arasındaki **nedensel ilişki grafiğini** oluşturmaktır <sup>18</sup>. Bu grafikte her bir zaman serisi bir düğüm (node) olarak temsil edilir ve **yönlü bir ok (kenar)**, bir seriden diğerine doğru bir nedensel etkiyi gösterir <sup>18</sup>. Grafikteki  $i$  to  $j$  yönündeki bir ok, “serisi  $i$ ’nin serisi  $j$  üzerinde nedensel etkisi vardır” anlamına gelir. Ayrıca her ok üzerinde, bu etkinin gerçekleştiği **zaman gecikmesi (lag)** belirtilir <sup>18</sup>. Örneğin bir ok üzerinde  $\tau=2$  yazması,  $i$  serisindeki değerlerin 2 zaman birimi sonra  $j$  serisinin değerlerini etkilediğini gösterir. Sonuç olarak elde etmek istediğimiz çıktı,  $G=(V, E)$  şeklinde bir **zamansal nedensellik grafiğidir**;  $V$  tüm zaman serilerinin kümesi (düğümler),  $E$  ise yönlü nedensel bağlantıların kümesidir

<sup>18</sup>.

## Yöntem: CausalFormer Modeli

Belirtilen sorunu çözmek için Lingbai Kong ve arkadaşları, **CausalFormer** adını verdikleri yeni bir derin öğrenme modeli önermektedir <sup>19</sup>. CausalFormer, **yorumlanabilir bir Transformer mimarisi** üzerine kuruludur ve iki ana bileşenden oluşur <sup>19</sup>:

- **(i) Nedensellik-farkında Transformer modeli (causality-aware transformer):** Zaman serisi verisi üzerinde bir tahmin görevi öğrenerek, verinin içindeki olası nedensel etkileşimleri gizli temsil olarak öğrenir. Bu modelin içine özel olarak tasarlanmış bir **çoklu-çekirdekli nedensel konvolüsyon** katmanı eklenmiştir; bu sayede her bir girdi serisi, zaman boyutu boyunca (geçmişten bugüne) **zamansal öncelik kısıtına uygun biçimde** özetlenir ve nedensel ilişkileri yakalayabilecek bir temsil haline getirilir <sup>19</sup>. Transformer yapısı, çoklu-başlık dikkat mekanizması ile birlikte kullanılarak seriler arası karmaşık ilişkilerin öğrenilmesini sağlar (aşağıda detaylandırılacaktır).
- **(ii) Ayırıştırma tabanlı nedensellik dedektörü (decomposition-based causality detector):** Eğitilmiş transformer modelinin iç yapısını **küresel (global) düzeyde yorumlayarak** nedensel ilişkileri ortaya çıkarır. Bu dedektör, klasik **katman-bazlı önem yayılımı** yöntemini (Layer-wise Relevance Propagation, LRP) regresyon problemine uyarlayan **Regresyon İlişki Yayılımı (Regression Relevance Propagation, RRP)** tekniğini içerir <sup>20</sup>. RRP sayesinde eğitilmiş modelin her katmanındaki çıktılara hangi girdilerin ne ölçüde katkı yaptığını izlemek mümkün olur. CausalFormer'ın nedensellik dedektörü, modelin çıktısını adım adım geriye doğru ayırıştırarak her bir olası etki-tepki çifti için bir **nedensellik skoru** hesaplar ve en sonunda tüm zaman serileri için nedensel ilişki grafiğini kurar <sup>20</sup>.

Yukarıdaki iki bileşen birlikte çalışarak, CausalFormer'ın **tüm model yapısını yorumlanabilir** hale getirdiğini vurgulayalım. Böylece sadece konvolüsyon veya dikkat ağırlıkları gibi lokal parametreler değil, modeldeki **bütün parametrelerin ve hesaplama yollarının** nedensel analize katkısı değerlendirilmektedir. Bu sayede, CausalFormer önceki yaklaşımların tespit edemediği **öz-nedensellik** (bir serinin kendi geleceğine etkisi) ve **eşzamanlı nedensellik** (aynı anda görülen neden-sonuç) gibi ilişkileri dahi kapsayacak şekilde **daha kapsamlı bir nedensel keşif** yapabilir <sup>21</sup>. Yazarların ifadeleriyle, CausalFormer “eğitilmiş derin öğrenme modelinin tüm yapısını yorumlayarak zaman serilerinin **tam nedensel ilişkilerini** öğrenmeyi” hedeflemektedir <sup>22</sup> <sup>23</sup>.

Aşağıda, önce CausalFormer modelinin mimarisini (Nedensellik-farkında Transformer) teknik detaylarıyla açıklayacağız. Ardından, bu modelin nasıl ayırıştırılarak nedensel skorlar elde edildiğini (Nedensellik Dedektörü ve RRP yöntemi) anlatacağız. Son bölümde deneysel sonuçlar ve modelin performansı ele alınacaktır.

### 4.1 Nedensellik-Farkında Transformer Mimarisi

CausalFormer'ın birinci ana bileşeni olan **nedensellik-farkında transformer**, temel olarak bir **çoklu değişkenli zaman serisi tahmin modelidir**. Klasik bir transformer mimarisine bazı özel katmanlar eklenerek uyarlanmıştır: modelde sırasıyla bir **Zaman Serisi Gömme Katmanı**, bir veya daha fazla **Çoklu-Çekirdekli Nedensel Konvolüsyon Bloku**, bir **Çok Değişkenli Nedensel Dikkat Bloğu**, ardından standart bir **İleri Besleme (feed-forward) katmanı** ve son olarak bir **Çıktı katmanı** bulunur <sup>24</sup> <sup>25</sup>. Modelin eğitimi, verilen çoklu zaman serisi girişinden yine aynı serilerin gelecek değerlerini tahmin etmek üzere kurgulanır (yani bir sonraki zaman adımını tahmin görevi) <sup>26</sup> <sup>27</sup>. Bu tahmin görevi sayesinde model, zaman serileri arasındaki ilişkileri öğrenmeye çalışır.

**4.1.1 Zaman Serisi Gömme Katmanı (Embedding):** Gömme katmanı, her bir zaman serisinin verisini yüksek boyutlu bir uzaya yansıtarak modelin sonraki katmanlarında kullanılmak üzere temsil özellikleri çıkarır <sup>28</sup>. Örneğin,  $x_i$  olarak ifade ettiğimiz  $i$  numaralı zaman serisinin son  $T$  zaman adımındaki verileri bir girdi vektörü olarak alınırsa, bu katman her bir seriyi  $d$  boyutlu bir vektöre dönüştürür. Bu işlem tipik olarak bir **doğrusal katman** ile yapılır:

$$(2) \mathbf{e}_i = W^{(emb)} \mathbf{x}_i + b^{(emb)}.$$

Burada  $\mathbf{x}_i$  serinin ham veri vektörünü (örneğin son  $T$  gözlemi) temsil eder,  $\mathbf{e}_i$  ise elde edilen **gömme vektörüdür** <sup>28</sup>.  $W^{(emb)}$  ve  $b^{(emb)}$  öğrenilebilir parametrelerdir. Gömme katmanının kritik bir noktası, bu çıktı  $\mathbf{e}_i$  vektörünün **sadece Transformer'ın dikkat mekanizmasındaki Sorgu (Query) ve Anahtar (Key) hesaplamalarında kullanılmasıdır**; buna karşın **Değer (Value)** kısmı için doğrudan ham zaman serisi verisinden hesaplanan konvolüsyon sonuçları kullanılacaktır <sup>29</sup>. Bunun sebebi şudur: gömme işlemi zaman boyutunu karıştırıp bilgiyi yoğunlaştırırken, Değer kısmının zaman sırasını koruması gerekir. Aksi takdirde model, nedensellik kısıtını (nedenin etkiden önce gelmesi kuralını) ihlal ederek geleceğe ait bilgiyi dikkate alabilir. Dolayısıyla **Temporal Priority (Zamansal Öncelik) kısıtını** sağlamak için, dikkat mekanizmasında Query ve Key daha soyut temsil alanında etkileşime girerken, Value doğrudan zaman sıralı özellikleri taşır <sup>29</sup>.

**4.1.2 Çoklu-Çekirdekli Nedensel Konvolüsyon:** Bu özel konvolüsyon katmanı, her bir zaman serisinin geçmiş (ve mevcut) değerlerini kullanarak, belirli bir hedef seriyi tahmin etmek üzere bir özet çıkarır. **CausalFormer'in yenilikçi yönlerinden biri**, her bir potansiyel *neden-sonuç* çifti için ayrı bir konvolüsyon çekirdeği öğrenmesidir <sup>30</sup>. Teknik olarak, modelde üç boyutlu bir konvolüsyon ağırlık tensörü  $\mathbf{K}$  tanımlanır. Bu  $\mathbf{K}$  tensörünün boyutları  $(N, N, L)$  şeklindedir <sup>31</sup>. Burada  $N$  giriş (neden) serisi sayısını,  $N$  tahmin edilecek (sonuç) serisi sayısını,  $L$  ise konvolüsyonun zaman penceresinin uzunluğunu (kullanılan gecikme adımı sayısını) temsil eder. Yani  $\mathbf{K}_{i,j}$  *dilimindeki boyut  $L$  uzunluğundaki vektör, " $i$  serisinden  $j$  serisine  $L$  uzunluğunda bir gecikme penceresiyle nedensel etki ağırlıkları" anlamına gelir. Bir başka ifadeyle,  $\mathbf{K}$  bir **öğrenilebilir filtre** olup,  $i$  serisinin geçmiş  $L$  değeri üzerinden  $j$  serisinin şimdiki değerine ne kadar katkı yapacağını hesaplar.*

Konvolüsyon işlemi şu şekilde gerçekleşir: model, her hedef  $j$  serisi ve her olası kaynak  $i$  serisi için bu  $\mathbf{K}_{i,j}$  *filtrelerini uygular. Örneğin,  $i$  serisinin  $t$  anına kadarki verilerine bakalım. Hesaplamanın nedensel (gecikmeli) olması için,  $t$  anındaki tahmini üretirken  $i$  serisinin yalnızca  $t$ 'den önceki değerleri kullanılmalıdır. Bunu sağlamak adına, giriş veri dizisinin başına  $L$  uzunluklu bir sıfır vektörü ile dolgu (padding) yapılır <sup>32</sup>. Şöyle ki,  $x_i$  serisinin verilerini  $[0, \dots, 0, x_i(0), x_i(1), \dots, x_i(t-1)]$  şeklinde sıfırlarla genişletiyoruz (en başa  $L-1$  adet sıfır eklenir) <sup>32</sup>. Ardından,  $\mathbf{K}$  filtresi bu doldurulmuş veri üzerinde evrişim (kaydırmalı iç çarpım) ile uygulanır. (3) numaralı denklem bu işlemi göstermektedir:*

$$(3) z_{i,j}(t) = \frac{1}{Z} \sum_{\tau=0}^{L-1} \mathbf{K}_{i,j}[\tau] \cdot x_i(t-1-\tau).$$

Burada  $z_{i,j}(t)$ ,  $i$  serisinin bilgisi kullanılarak  $j$  serisinin  $t$  anı için elde edilen konvolüsyon sonucudur.  $\mathbf{K}_{i,j}[\tau]$  ile filtrenin  $\tau$  gecikmesine ait ağırlığını,  $x_i(t-1-\tau)$  ile de  $i$  serisinin ilgili geçmiş değerini gösteriyoruz. Eşitliğin sağ tarafında  $Z$  ile belirtilen normalizasyon faktörü ise konvolüsyon penceresinin veriye katkı yapan gerçek eleman sayısıdır (pencere veri dizisinin başında kaç tane sıfır olduğuna bağlı olarak değişir) <sup>33</sup>. Böylece kenarlarda daha az veri olduğunda ölçekleme yapılarak tutarlı bir katkı hesaplanır <sup>34</sup>. Elde edilen  $z_{i,j}(t)$  değeri, " **$i$  serisinin  $j$  serisine  $t$  anında yaptığı nedensel katkı**" olarak yorumlanabilir <sup>35</sup>. Tüm  $i$  serileri için bu işlemler yapıldığında,

her bir  $j$  hedefi için  $\mathbf{z}_{\cdot j}(t) = [z(t)]$  şeklinde bir vektör oluşur; bu vektör  $j$  serisine o anda etki eden bütün diğer serilerin anlık katkılarını içermektedir <sup>35</sup>.  $\mathbf{z}_{\cdot j}(t)$ ,  $\mathbf{z}_{\cdot 2,j}(t)$ , ...,  $\mathbf{z}_{\cdot N,j}$

Bir önemli nokta da, **öz-nedensellik (self-causation)** durumunun doğru ele alınmasıdır. Eğer  $i = j$  ise, yani bir serinin kendi geleceğini etkileme potansiyelini yakalamak istiyorsak, yukarıdaki konvolüsyon işlemi özel bir şekilde uygulanır. Model, bir serinin kendi konvolüsyon sonucunu **bir zaman adımı kaydırarak** kullanır <sup>36</sup>. Bu, (3) nolu denklemden gelen  $\tilde{z}_{\cdot i,j}(t)$  değerinin  $t$  anındaki tahmine doğrudan girmemesi, bunun yerine  $t+1$  gibi geleceğe kaydırılması anlamına gelir. Makaledeki (4) nolu denklem bu işlemi sembolik olarak ifade etmektedir:

$$(4) \quad \tilde{z}_{\cdot i,j}(t) = z(t-1).$$

Bu şekilde, bir serinin kendi değerinin tahminine katkısı hesaplanırken model **asla o serinin gerçek güncel değerini kullanmaz**, her zaman bir önceki zamana kaydırır <sup>36</sup>. Bunun amacı, model eğitimi sırasında "kendi kendini açıklama" durumunda hileli bir şekilde gerçek değeri kopyalamayı engellemektir. Böylece öz-nedensellik öğrenilirken, model bir serinin önceki değerlerinden şimdiki değerine olan etkiyi gerçekten yakalamak zorunda kalır (ground truth değeri açıklamaya dahil edilmediği için) <sup>37</sup>.

CausalFormer modelinde bu **çoklu-çekirdekli nedensel konvolüsyon blokları** birden fazla kullanılabilir ya da farklı çekirdek grupları halinde organize edilebilir. Her bir blok, tüm seriler arası etkileşimleri bu şekilde hesaplar ve sonuçlarını bir sonraki katmana iletir. Yazarlar, birden fazla konvolüsyon çekirdeği kullanmanın ("multi-kernel") her bir seri çiftine ait bağımsız veri akışları kurarak nedensel temsilin daha ince ayrıntılarını yakalamaya yardımcı olduğunu belirtmektedir <sup>30</sup> <sup>38</sup>. Yani, bir seri çifti arasındaki ilişki belki farklı gecikme pencerelerinde veya farklı dinamiklerde etkili olabilir; çoklu çekirdekler bu farklı etki paternlerini ayrı ayrı öğrenip sonradan birleştirebilir.

**4.1.3 Çok Değişkenli Nedensel Dikkat (Multi-variate Causal Attention):** Transformer mimarisinin kalbinde yer alan **dikkat mekanizması (attention)**, CausalFormer'da *nedensel ilişkilere uygun şekilde* yeniden ele alınmıştır. Bu dikkat bloğu, birden fazla zaman serisi arasındaki **daha karmaşık ve dolaylı nedensel bağıntıları** yakalamayı amaçlar.

Klasik bir çoklu-başlıklı (multi-head) dikkat mekanizmasında her bir başlık için **Sorgu (Query)**, **Anahtar (Key)** ve **Değer (Value)** matrisleri hesaplanır. CausalFormer'da Sorgu ve Anahtar, zaman serilerinin **gömme vektörlerinden** elde edilirken, Değer matrisi **konvolüsyon bloklarının sonuçlarından** elde edilir. Formül olarak:

$$(5) \quad Q = W^Q E + b^Q, \quad K = W^K E + b^K, \quad V = Z.$$

Burada  $E$  matrisi, tüm serilerin embedding vektörlerini bir araya getirir;  $W^Q$ ,  $W^K$  matrisleri ve  $b^Q$ ,  $b^K$  vektörleri öğrenilebilir parametrelerdir.  $Z$  ise bir önceki konvolüsyon aşamasından gelen (tüm seri çiftleri için) çıktı matrisidir <sup>39</sup>. Dikkat edersek,  $V$  için herhangi bir ağırlıkla dönüşüm yapılmamış, doğrudan  $Z$  kullanılmıştır. Bu, daha önce belirtildiği gibi Value'nin zaman bilgisini olduğu gibi taşıması gerektiğindendir (embedding yapılırsa zaman sırası karışır).  $Q$  ve  $K$  ise seriler arası karşılaştırmayı daha soyut bir düzlemde yapabilmek için projeksiyondan geçirilmiştir.

Dikkat hesaplaması,  $Q$  ve  $K$  matrislerinin birbiriyle etkileşiminden elde edilen benzerlik skorlarının  $V$  üzerindeki ağırlıklı ortalaması ile yapılır. CausalFormer'da bu mekanizma aşağıdaki şekilde modifiye edilmiştir:

$$(6) \text{Softmax}\left(\frac{QK^T}{\tau}\right), \text{Attention\_Output} = A \cdot V$$

Burada  $QK^T$  ifadesi geleneksel olarak Sorgu ve Anahtar matrislerinin benzerlik (dikkat) skorlarını verir.  $M$ , modeli eğitirken öğrenilen bir **mask matrisidir**; bu maske, dikkat matrisinin seyrekliğini ayarlamak için kullanılır <sup>40</sup>. Başka bir deyişle  $M$ , bazı ilişkilerin ağırlığını azaltmak veya sıfıra yakınlamak için optimize edilen ekstra bir parametredir, bu sayede elde edilecek nedensel grafiğin gereksiz karmaşıklığı önlenir (dikkat bağlantılarının **seyrekleştirilmesi** sağlanır) <sup>40</sup>.  $\tau$  ise sıcaklık (temperature) hiperparametresidir; softmax hesaplamasında ölçeklendirme yaparak dikkat dağılımının keskinliğini kontrol eder <sup>40</sup>. Bu formülasyondan elde edilen  $A$  matrisi, boyutu  $N \times N$  olan **dikkat matrisidir** <sup>41</sup>.  $A$ 'nın  $i, j$  elemanı, kabaca " $i$  serisine ait özelliklerin  $j$  serisinin tahminine ne kadar katkıda bulunduğunu" gösterir şekilde yorumlanabilir.

Her bir dikkat başlığı (head) için bu işlem yapılır ve sonuçları  $V$  ile çarpılarak bir çıktı üretilir:  $\text{Attention\_Output} = A \cdot V$ . Ardından çoklu-başlık yapıyı elde etmek için,  $h$  adet başlığın çıktılarını yan yana birleştirip bir çıktı ağırlığı matrisi  $W^O$  ile yeniden ağırlıklandırarak birleştiririz <sup>42</sup>:

$$(7) H_{\text{out}} = [\text{head}_1, \text{head}_2, \dots, \text{head}_h] \cdot W^O$$

Burada köşeli parantez, başlık çıktı matrislerinin yan yana konkatene edilmesini göstermektedir <sup>42</sup>.  $W^O$  bu birleştirilmiş matrisi tekrar orijinal boyuta dönüştüren bir öğrenilebilir ağırlıktır. Böylece **Çoklu-Değişkenli Nedensel Dikkat Bloğu**,  $N$  serisinin konvolüsyonla çıkarılmış özniteliklerini alıp, çeşitli proje alanlarında etkileştirerek her bir seri için diğer serilerden gelen karmaşık etkileri hesaplamış olur. Bu bloğun çıktısı, boyut olarak yine  $N$  serilik bir özet (her seri için bir özellik vektörü) olacaktır.

**4.1.4 İleri Besleme Katmanı:** Transformer mimarisinin standart bir parçası olan ileri besleme (feed-forward) katmanı, dikkat mekanizması sonrası elde edilen çıktıları doğrusal ve doğrusal olmayan dönüşümler uygulayarak modeli güçlendirir. CausalFormer'da ileri besleme katmanı iki adet tam bağlı (fully-connected) katmandan oluşur; bu iki katman arasına bir **Leaky ReLU** aktivasyon fonksiyonu yerleştirilmiştir <sup>43</sup>. Yapısı şu şekildedir:

$$(8) \text{FFN}(x) = W_2 \cdot \sigma(W_1 x + b_1) + b_2$$

burada  $\sigma$  Leaky ReLU aktivasyonunu temsil eder. İlk katmanın ağırlığı  $W_1$  boyutu genişletir (ör.  $d$ 'den  $d_{\text{ff}}$ 'ye çıkarır), ikinci katman  $W_2$  ise tekrar orijinal boyuta indirger <sup>44</sup>. Bu katman, modele doğrusal olmayan bir dönüşüm ekleyerek dikkat mekanizmasıyla elde edilen nedensel temsil gücünü artırır <sup>45</sup>.

**4.1.5 Çıktı Katmanı ve Kayıp Fonksiyonu:** Transformer bloğunun en sonunda, her bir serinin tahminini üreten bir **çıktı katmanı** bulunur. Bu, dikkat+FFN çıkışlarına uygulanan basit bir tam bağlı (linear) katmandır. Sonuçta model, her zaman adımı için  $N$  boyutlu bir tahmin vektörü (tüm serilerin bir sonraki değer tahminleri) üretir:  $\hat{x}(t) = W^{\text{out}} H_{\text{out}} + b^{\text{out}}$ . Modelin eğitimi için standart **Ortalama Kare Hatası (Mean Squared Error, MSE)** kayıp fonksiyonu kullanılır <sup>46</sup>. Yani model, tahmin ettiği değerler ile gerçek gözlemler arasındaki MSE'yi minimize edecek şekilde öğrenir. Ancak burada iki önemli eğitim detayı vardır:

- **İlk zaman adımı atılması:** Model eğitilirken, her serinin kendi geçmişinden faydalananak geleceğini tahmin etmesini sağlamak istiyoruz. Özellikle öz-nedensellik için modelin adil değerlendirilmesi amacıyla, her serinin ilk zaman adımıdaki tahmini eğitim kaybına **dahil**

**edilmiyor** <sup>47</sup> . Çünkü konvolüsyon bloğunda kendi kendine etki (self-convolution) sonucu bir adım kaydırıldığı için, ilk adımdaki  $\hat{x}_{i(t_0)}$  tahmini,  $x_i$  serisinin geçmiş verisi olmadığından tamamen 0 pad'den gelir (kendi serisine dayanmaz). Bu nedenle, her serinin  $t_0$  anındaki hatası hesaplamaya katılmaz, böylece tüm seriler için adil bir karşılaştırma sağlanır <sup>46</sup> .

- **Seyreklik cezaları:** Makalenin önerdiği modelde, öğrenilen nedensel ilişkilerin **mümkün olduğunca az ve öz olması** istenir. Aşırı karmaşık, her serinin her biriyle bağlantılı olduğu bir grafik yerine, gerçekten anlamlı ilişkileri bulmak amaçlanır. Bu nedenle kayıp fonksiyonuna, modelin nedensellik parametrelerini seyrekleştirmeye yönelik ceza terimleri eklenmiştir <sup>48</sup> . Özellikle, **konvolüsyon çekirdekleri** ( $\mathbf{K}$ ) tensöründeki ağırlıklar ve **dikkat maskesi matrisi**  $M$  üzerine norm tabanlı bir regularizasyon uygulanır <sup>48</sup> . Formül olarak, toplam kayıp şu şekilde ifade edilebilir:

$$(9) \quad L_{\text{toplam}} = \frac{1}{T-1} \sum_{t=t_0+1}^{t_0+T-1} \|\hat{\mathbf{x}}(t) - \mathbf{x}(t)\|_2^2 + \gamma \|\mathbf{K}\|_1 + \zeta \|M\|_1$$

Yukarıdaki denklemde ilk terim, tüm seriler için (ilk zaman adımı hariç) MSE tahmin hatasını temsil eder. Sonraki terimler ise  $\gamma$  ve  $\zeta$  katsayılarıyla ölçeklendirilmiş  $L_1$  norm cezalarıdır;  $\|\mathbf{K}\|_1$  tüm konvolüsyon çekirdek ağırlıklarının toplam mutlak değerini,  $\|M\|_1$  ise mask matrisinin elemanlarının toplam mutlak değerini ifade eder.  $L_1$  normunun kullanılması, **seyrek çözümleri teşvik etmesi** ile bilinir; yani model gereksiz bağlantı ağırlıklarını sıfıra yaklaştırmaya çalışır. Sonuç olarak, bu regularizasyon terimleri sayesinde CausalFormer, öğrenirken gereksiz nedensel bağlantıları baskılar ve daha sade bir ilişki yapısı elde etmeye yönelir <sup>49</sup> .

Özetlemek gerekirse, Causality-aware Transformer bileşeni, sıradan bir transformer'ın zaman serisi tahmini yapacak şekilde uyarlanmış ve nedenselliği daha açık temsil edecek özel katmanlarla zenginleştirilmiş halidir. Bu model eğitildikten sonra, içindeki parametrelerde ve ara değerlerde bir şekilde "gizli" olarak zaman serileri arası nedensel ilişkilerin izleri bulunacaktır. **Ancak bu ilişkileri doğrudan parametre değerlerinden çıkarmak hala kolay değildir.** Örneğin bir dikkat matrisi  $A$  elde ettik, bunun büyük değerleri nedensel ilişki göstergesi midir? Kısmen evet, ama tamamen değil; çünkü modelin sonraki katmanları (FFN ve çıktı) bu değerlere müdahale etmektedir. İşte bu noktada devreye modelin ikinci bileşeni giriyor: **ayırıştırma tabanlı nedensellik dedektörü**. Bu dedektör, eğitilen modelin tamamını girdi-çıkı katkılarını açısından analiz ederek gerçekten hangi serinin hangi seriyi etkilediğini hesaplar. Şimdi bu bileşenin detaylarına geçelim.

## 4.2 Ayırıştırma Tabanlı Nedensellik Dedektörü

CausalFormer'ın ikinci ana bileşeni, eğitilmiş nedensellik-farkında transformer modelini alıp, onun iç hesaplamalarını geriye doğru çözümleyerek **nedensel ilişki skorlarını** çıkaran kısımdır <sup>50</sup> . Bu bileşenin temel yeniliği, **Regresyon Relevance Propagation (RRP)** adı verilen yorumlama tekniğidir. RRP, sınıflandırma modellerini açıklamak için kullanılan **katman bazlı ilgililik yayılımı (Layer-wise Relevance Propagation, LRP)** yöntemini regresyon problemine genelleştirir <sup>20</sup> . LRP'den kısaca bahsedersen, bu yöntem bir sinir ağının çıktısını, girişindeki her bir ögenin ne kadar katkı yaptığını hesaplamak üzere katmandan katmana **geriye doğru dağıtır**. İlk olarak görüntü sınıflandırma gibi alanlarda her bir pikselin sınıfa olan katkısını hesaplamak için geliştirilmiştir.

**4.2.1 Regresyon için İlgililik Yayılımı (Regression Relevance Propagation - RRP):** RRP'yi anlamak için öncelikle LRP'nin klasik formülasyonuna bakalım. **Figür 4**, LRP'nin basit bir ağ üzerindeki işleyişini şematize etmektedir <sup>51</sup> . Bir ağ tek bir gözlem  $\mathbf{x}$  üzerinde bir çıktı ürettiğinde (örneğin bir sınıf olasılığı), LRP bu çıktıyı  $R^{(L)}$  olarak ele alır (en son katmandaki relevans skoru) ve bunu giriş katmanına kadar paylaştırır. Her bir ara katmanda  $R_i^{(l)}$  terimleri, **o katmandaki  $i$  birimin ne**

**kadar “ilgili” (relevant) olduğunu**, yani çıktıya ne kadar katkıda bulunduğunu gösterir <sup>52</sup>. LRP’nin temel ilkesi **korunum (conservation)** ilkesidir: Her katmanda paylaştırılan relevans skorlarının toplamı, bir üst katmandaki toplam skora eşit olmalıdır <sup>52</sup>. Formülle ifade edilirse,  $l$  katmanda yer alan  $i$  nci nöronun skoru  $R_j^{(l)}$  ve  $l+1$  katmandaki  $j$  nci nöronun skoru  $R_j^{(l+1)}$  ise, aşağıdaki eşitlik sağlanır:

$$(10) \quad R_j^{(l+1)} = \sum_{i \in \text{Katman } l} R_i^{(l)}$$

Burada  $R_j^{(l)}$  çıktının kendisini (örneğin bir sınıf olasılığı değeri) temsilleyebilir. LRP algoritması, çıktındaki bu skoru (veya her bir çıktı nöronu için skorları) **geriye doğru dağıtarak** giriş özneteliklerine kadar indirger. Elde edilen  $R_k^{(0)}$  değerleri, girdinin her bir özelliğinin (örneğin her bir pikselin) ne kadar katkı yaptığını gösterir. **Pozitif ve büyük bir relevans skoru**, ilgili girdinin çıktıyı artırıcı yönde önemli bir etkiye sahip olduğunu (yani bir nedensel faktör olabileceğini) gösterir. Özellikle zaman serilerinde Granger nedenselliği açısından düşünürsek, **bir girdi serisinin yüksek pozitif relevans skoru alması, onun tahmin edilen hedef seride nedensel etkisi olduğuna işaret eder** <sup>53</sup>.

Klasik LRP sınıflandırma ağları için tanımlandığından, çıktı olasılıkları üzerinden çalışır. Bu durumda genellikle *referans (köken) noktası* denilen bir kavram kullanılır: Sınıflandırmada, belirli bir girdinin önemli özelliklerini anlamak için onu “anlamsız” bir örnek ile karşılaştırırız. Örneğin bir görüntüde, hiç bilgi içermeyen boş bir resim referans alınabilir; bir girdi görüntü ile referansı arasındaki farklar hangi piksellerin sonucu pozitif yaptığını gösterir. Teknik olarak LRP’nin yayılım kuralı, **Taylor serisi açılımı** ile de ilişkilendirilir. Derin Taylor Ayrıştırması adı verilen yöntem, bir ağın çıktı fonksiyonunu ilk dereceden Taylor açılımı ile yaklaşık olarak relevans skorlarını bu açılımdaki terimlerden elde eder <sup>54</sup>. Tek katmanlı bir ağ için, bir  $f(\mathbf{x})$  çıktı fonksiyonunu  $f(\mathbf{x}_0)$  çevresinde birinci dereceden yaklaşırsak:

$$(11) \quad f(\mathbf{x}) \approx f(\mathbf{x}_0) + \sum_i (x_i - x_{i0}) \frac{\partial f}{\partial x_i} \Big|_{\mathbf{x}_0}$$

Burada  $f(\mathbf{x}_0)$  *referans (kök) noktasıdır*,  $(x_i - x_{i0})$  numaralı girdinin sonuca katkısını gösterir; dolayısıyla bunlar  $\frac{\partial f}{\partial x_i}$  girdiyle referans arasındaki farktır. Bu formülde toplam içindeki her bir terim  $(x_i - x_{i0}) \frac{\partial f}{\partial x_i}$  **relevans skorları** olarak alınır <sup>55</sup>. Klasik LRP yaklaşımı bu fikri çok katmanlı ağlara uygular, her katmanı lineer parçalar olarak ele alıp referans noktasını genellikle sıfır aktivasyon veya nötr girdi olarak seçer.

Ancak, **CausalFormer gibi bir regresyon modelinde** doğrudan olasılık gibi bir çıktı yoktur; modelin çıktısı sürekli bir değer (tahmin edilen veri değeri) olduğundan, sıfır değerini referans almak her zaman anlamlı olmaz. Özellikle, eğer tahmin etmeye çalıştığımız gerçek değer (ground truth) zaten sıfıra yakınsa, referans noktası seçimi belirsizleşir <sup>56</sup>. Yazarlar bu sorunu aşmak için RRP yönteminde farklı bir strateji izliyor: **Model çıktısını kademeli olarak sıfıra düşürecek bir referans noktası arıyorlar**. Bunu sağlamak için, yorumlanacak girdi vektörünü (zamansal veri örneğini) orijinden (sıfır vektöründen) itibaren ölçeklendiren bir çizgi boyunca hareket ettiriyorlar <sup>57</sup> <sup>58</sup>. Başka bir deyişle,  $f(\mathbf{x})$  değerini sıfıra indirecek  $f(\mathbf{x}_0)$  noktasını (yani “boş” tahmin üreten girdiyi) bulmaya çalışıyorlar.

**Şekil 5**, bu kök noktanın en yakın nerede olduğunun geometrik ararışını göstermektedir <sup>59</sup>. Tek katmanlı doğrusal bir ağ için,  $w$  ağırlıkları ve  $b$  bias’ı ile  $y = w \cdot x + b$  çıkışı alındığında,  $y=0$  düzlemi  $w \cdot x + b = 0$  denklemiyle verilir. Girdi vektörü  $\mathbf{x}$ ’ten orijin yönüne bir doğru çizdiğimizde, bu doğrunun denklemi  $\mathbf{x}_0 + \alpha \cdot \mathbf{d}$  (burada  $\mathbf{d}$  yön vektörü,  $\alpha$  ise ölçek parametresi) olabilir. Bu ikisinin kesişimini bularak  $\mathbf{x}_0$  (kök nokta) elde edilir <sup>58</sup>. Denklem formunda:



$$(12) \quad w_i (x_{0,i} - \alpha d_i) + b = 0,$$

bu denklemi  $\alpha$  için çözdüğümüzde ve (11) nolu Taylor açılımından gelen terimlerle birleştirdiğimizde, her bir girdi için relevans skoru formülü güncellenir <sup>60</sup>. Sonuç olarak tek katman için bulunan:

$$(13) \quad R_i = \frac{(x_i - x_{0,i}), w_i \sum_k (x_k - x_{0,k}), w_k}{R_{\text{out}}},$$

ifadesi,  $i$  nci girdinin relevans skorunu vermektedir (burada  $R_{\text{out}}$  çıkış katmanının toplam skorudur, genelde  $y$  nin kendisi). Bu formül, LRP literatüründe "**z-kuralı**" olarak bilinen özel bir durumla ilişkilidir. Z-kuralı, relevans skorlarını hesaplamak için referans noktasını orijine bağlayan doğru yönünü seçer <sup>61</sup>. Basitçe söylemek gerekirse,  $d_i = x_i$  alınır (girdi vektörünün kendisi yön vektörü olur). Bu durumda yukarıdaki formül,  $R_i = \frac{x_i w_i \sum_k x_k w_k}{R_{\text{out}}}$  şeklini alır <sup>61</sup>. Yani her bir girdi özelliği, ağırlıklı ortalama oranında çıkış skorunu paylaşıyor demektir.

CausalFormer yazarları bu formülü, **bias teriminin de etkisini** hesaba katacak şekilde genişletiyor. Normalde (13) formülünde payda  $\sum_k x_k w_k$  idi; buna bias  $b$ 'yi de ekliyorlar <sup>62</sup>. Böylece:

$$(15) \quad R_i = \frac{x_i w_i \sum_k x_k w_k + b}{R_{\text{out}}},$$

$$(16) \quad R_b = \frac{b}{\sum_k x_k w_k + b} R_{\text{out}}.$$

Burada (15) nolu denklem  $i$  nci girdinin relevans skorunu, (16) ise bias teriminin relevans skorunu veriyor <sup>63</sup>. Dikkat edilirse, artık relevans skorlarının toplamı tam olarak  $R_{\text{out}}$  etmiyor; çünkü bias terimi çıkış skorunun bir kısmını "alıp götürüyor". Yani  $\sum_i R_i + R_b = R_{\text{out}}$  olacak,  $R_b$  da genelde pozitif çıkacaktır. Bu durum, **LRP'nin katman bazında korunum koşulunu ihlal ediyor** gibi görünse de aslında kasti bir tercih <sup>64</sup>. Zira eğer bias terimi çok büyükse, bu modelde girdi verilerinde açıklanamayan bir sabit etkinin olduğunu gösterir. Bias'a yüksek bir relevans skoru düşmesi demek, model çıktısının önemli bir kısmının veri dışı bir sabit faktörden geldiği anlamına gelir. Bu durumda yazarlar diyor ki: "*Bias yüksek relevans aldıysa, ilgili nedensel ilişkinin gücünü biraz düşürmek aslında makul olabilir.*" Çünkü bir kısmı veriyle açıklanmıyor (bias ile açıklanıyor). Bu nedenle (15)-(16) denklemlerindeki bias ayrımı, katman korunumunu bozsa da **nedensel analizin doğruluğu için faydalı** kabul edilmiş <sup>64</sup>. Bias'ın yüksek etkili olduğu durumlarda, model otomatik olarak o bağlantının nedensel skorunu biraz aşağı çekiyor (bias skoru ne kadar büyükse, girdi skorundan o kadar azaltıyor) <sup>65</sup>.

Bu aşamadan sonra son bir adım kalıyor: Formülü, tek katmandan tüm ağa genellemek. Yazarlar, (15) nolu denklemdeki  $x_i w_i$  terimini, genel bir ağ katmanı için *o katmanın  $i$  nci nöronunun çıktısı*; türev terimini ise  *$j$  nci üst katman çıktısının  $i$  nci alt katman girişine göre türevi* ile değiştiriyorlar <sup>66</sup>. Böylece herhangi bir katman için genel bir relevans yayılım kuralı elde ediliyor. Bu kural, makalede (17) nolu denklemle verilmiş:

$$(17) \quad R^{(l)}_i = \frac{z^{(l)}_i \partial (y^{(l)}) / \partial (z^{(l)}_i) \sum_k z^{(l)}_k \partial (y^{(l)}) / \partial (z^{(l)}_k) R^{(l+1)}_j}{R^{(l+1)}_j}$$

Burada  $z^{(l)}_i$  ile  $i$ \$. katmandaki  $i$  numaralı birimin çıktısını,  $y^{(l)}$  ile de  $i$ \$. katmanın çıktı vektörünü (o birimin de dahil olduğu) ifade edebiliriz.  $R^{(l+1)}_j$  üst (sonraki) katmandaki  $j$ \$. birimin relevans skoru,  $R^{(l)}_i$  ise alt (önceki) katmandaki  $i$ \$. birimin aradığımız relevans skorudur. Formülün özü, önceki (üst) katmandaki  $j$  nöronun skorunu, o nörona  $i$  tarafından gelen katkının

oranına göre  $\$i$ 'ye paylaştırmaktır <sup>66</sup>. Bu formül, aşağıdaki **her türlü parametreli katmana** (doğrusal, konvolüsyon vb.) uygulanabilir durumda tasarlanmıştır <sup>67</sup>.

Ağın içinde parametre içermeyen işlemler de olabilir (örneğin matris çarpımı, aktivasyon fonksiyonu vs. – toplayıcı bir düğüm gibi). Bunlar için de yazarlar **parametresiz relevans yayılımı** adı verilen yöntemlerden yararlanıyor <sup>68</sup>. Örneğin bir matris çarpımında ( $R = P \times Q$  gibi,  $P$  ve  $Q$  iki girdi matrisi,  $R$  çıktı matrisi), relevans skoru hem  $P$  hem  $Q$  tarafına paylaştırılmalıdır. Makalede (18) nolu denklemde buna dair bir kural veriliyor:

$$(18) \quad R^{(l)} \cdot P = \frac{P \cdot (R^{(l+1)} \cdot Q^{\text{mathsf{T}}})}{P \cdot Q \cdot R^{(l+1)}}, \quad R^{(l)} \cdot Q = \frac{Q \cdot (P^{\text{mathsf{T}}} \cdot R^{(l+1)})}{P \cdot Q \cdot R^{(l+1)}}$$

Bu formülde  $P \cdot X$  iki matrisi element-wise (bileşen bazında) çarpımı,  $PQ$  ise normal matris çarpımını ifade etmektedir <sup>69</sup>. Detaya girmeden, özetle şunu belirtelim: bu tip kurallarla modelin içindeki her türlü hesaplama için relevans skorlarını girdi bileşenlerine kadar izlemek mümkün hale getirilmiştir.

RRP yönteminin sonuç olarak yaptığı şey şudur: Causality-aware Transformer modeline belli bir hedef değişken seçilerek giriş verisi verilir, model tahminini üretir; sonra bu tahmin sonucu tek bir seri (hedef seri) için **öne çıkarılır** (çünkü biz kimin o seriyi etkilediğini bulmak istiyoruz) ve bu çıktı skoru modelin tüm katmanlarından geriye doğru dağıtılır. Dağıtılırken her adımda yukarıdaki gibi kurallar uygulanır ve en sonunda modelin dikkat katmanındaki **dikkat matrisi  $SA$**  ve konvolüsyon katmanındaki **çekirdek tensörü  $SK$**  seviyesine kadar relevans skorları hesaplanır <sup>70</sup>. **Şekil 6 (a)**, bu sürecin genel hatlarını göstermektedir: Eğitilmiş transformer modeline bir veri kümesi verilir, hedef serinin tahmin çıktısı alınır; ardından RRP ile geriye gidilerek dikkat ve konvolüsyon parametrelerine kadar her bir bağlantı için relevans skorları elde edilir <sup>70</sup>.

RRP'nin uygulanmasında ufak bir ayrıntı: Her defasında **hangi serinin nedenlerini bulmak istediğimizi** belirlemeliyiz. Örneğin elimizde 5 zaman serisi varsa ve biz birinci seriye etki eden diğer serileri arıyorsak, RRP başlatılırken çıktı katmanındaki relevans skorları vektörü **one-hot vektör** olarak alınır (sadece 1. seriye karşılık gelen pozisyon 1, diğerleri 0) <sup>71</sup>. Bu sayede geri yayılım, özellikle birinci seriye odaklanarak yapılır ve diğer serilerin çıktıları karıştırılmaz. Bunu veri kümesindeki her seri için teker teker yaparak, her serinin muhtemel sebeplerini bulabiliriz.

**4.2.2 Gradyan Modülasyonu:** Yazarlar, saf relevans skorlarını daha da iyileştirmek için **gradyan bilgisini** de sürece dahil etmişlerdir. Gerekçeleri şu: Bir girdi özelliğinin model çıktısı üzerindeki etkisinin büyüklüğü sadece o anki aktivasyonuna (relevans ile ölçülen etkiye) değil, aynı zamanda model çıktısının o girdiye duyarlılığına (gradyanına) da bağlıdır <sup>72</sup>. Eğer bir girdi küçük bir değişiklik yaptığında çıktı büyük değişiyorsa, bu girdi nedensel olarak önemlidir. Bunu hesaba katmak için, bulunan relevans skorları, ilgili girdinin **gradyanının mutlak değeri** ile çarpılarak güncellenir <sup>72</sup>. Formülle:

$$(19) \quad C_{i^{(l)}} = \max(0, G_{i^{(l)}}) \times R_{i^{(l)}},$$

burada  $R_{i^{(l)}}$   $i$  nci girdinin (örneğin bir serinin belirli bir gecikmeyle ilgili bileşeninin) relevans skoru,  $G_{i^{(l)}}$  ise aynı girdinin ilgili çıktı üzerindeki gradyanıdır;  $\max(0, \cdot)$  ise negatif değerleri 0 yapan ReLU benzeri bir işlemdir <sup>73</sup>. Elde edilen  $C_{i^{(l)}}$  değeri **nedensel skor (causal score)** olarak adlandırılır. Tüm negatif gradyan katkıları sıfırlanır, sadece pozitif yönde (çıktıyı artıran yönde) etkiler tutulur <sup>74</sup>. Bu sayede, model çıktısını azaltan ilişkiler (negatif korelasyonlu etkiler) analizde elenmiş olur. Özellikle Granger nedenselliği çerçevesinde, bir serinin başka bir seriyi pozitif yönde etkilemesi tipik olarak “nedensel” sayılır; negatif etkileşimler ise farklı bir yorum gerektirebilir (örneğin inhibe edici

etki), fakat yazarlar bunları **non-causal (nedensel olmayan)** olarak değerlendirmeyi tercih etmişlerdir <sup>74</sup> . Dolayısıyla bu adım, yalnızca pozitif nedensel etkilere odaklanmayı sağlamaktadır.

**4.2.3 Nedensel Grafın Oluşturulması:** Son olarak, elde edilen nedensel skorları kullanarak zaman serileri arasındaki grafiği çıkarmak gerekir. Yukarıda anlattığımız süreç sonucunda, her bir hedef seri  $j$  için iki tür bilgi elde etmiş oluyoruz: (a) Dikkat matrisi  $A$  üzerinden gelen nedensel skorlar (hangi seri  $i$  dikkat mekanizmasında  $j$ 'ye ne kadar etki yaptı), (b) Konvolüsyon çekirdekleri üzerinden gelen skorlar (hangi seri  $i$ ,  $j$ 'yi hangi gecikmeyle etkiledi). Bu skorları **haritalamak** için yazarlar şöyle bir yaklaşım kullanıyor <sup>75</sup> <sup>76</sup> :

- Öncelikle embedding veya lineer projeksiyon katmanları gibi, seriler arası bilgi karıştırmayan katmanlar RRP sürecinde atlanmıştır. Zira bu katmanlar sadece tek bir serinin temsiline etki eder, dolayısıyla nedensel ilişkiler hakkında doğrudan bilgi taşımazlar <sup>75</sup> . Bizim ilgilendiğimiz skorlar, **dikkat matrisi  $A$**  ve **konvolüsyon çekirdek tensörü  $K$**  üzerinde hesaplanan skorlardır. Bunlar, hangi seriden hangi seriye potansiyel etki ve ne kadarlık gecikme olduğu hakkında ipucu verir.
- Diyelim hedef serimiz  $j$ . Dikkat matrisindeki skorlardan  $j$  ile ilgili olanları alalım: bu kabaca  $A$  matrisinin tüm satırlarının  $j$  sütununa ilişkin değerleri olacaktır (her  $i$  için bir skor). Bu değerler,  **$i$  serisinin  $j$ 'ye etkisinin "kümelerini" bulmak için kullanılır** <sup>77</sup> . **Yazarlar bu değerlerin dağılımını analiz etmek için  $k$ -ortalama (k-means) kümeleme algoritmasını kullanıyor. Yani  $A$  üzerindeki  $N$  adet  $i$  için  $j$  skoru,  $k$  adet sınıfa ayrılıyor** <sup>78</sup> . **Bu  $k$  hiperparametresi, grafın ne kadar yoğun (çok bağlantılı) olacağını kontrol ediyor; büyük  $k$  değerleri daha fazla küme (daha incelikli ayırım) demek ve bu genelde daha fazla nedensel ilişki çıkmasına yol açıyor (her bir sınıftan en yüksek olan alındığı için)** <sup>78</sup> . **Kümeleme sonucunda, hangi serilerin yüksek skorlu, hangilerinin düşük skorlu olduğu ortaya çıkar. En yüksek ortalamaya sahip küme,  $j$  serisine gerçekten etki eden serileri temsil eder olarak yorumlanır. Bu kümedeki serilerin  $A$  skorları en büyük olduğundan, hedef seriyi etkileyen aday nedenler\*\* olarak seçilirler** <sup>78</sup> . Özellikle en yüksek skora sahip sınıftaki serileri alıp, bunlar arasında da en belirgin birkaçını seçmek bir yaklaşımdır (ancak makale tüm sınıfı birden alıyor gibi anlaşıyor). Sonuçta her bir  $j$  için bir nedenler listesi çıkar.
- Bir seri  $i$ 'nin  $j$  üzerindeki etki olduğunu belirledikten sonra, sıra bu etkinin **gecikmesini (lag)** belirlemeye gelir. Bunun için konvolüsyon çekirdek tensöründen gelen skorlara bakılır <sup>79</sup> . Hatırlarsak,  $K_{i,j}$   $L$  boyutlu bir filtre idi ve  $z_{i,j}(t)$  hesaplanırken bunun her bir  $\tau$  gecikmesindeki ağırlığı çarpılıyordu. RRP ile biz  $K_{i,j}$ 'nin her bir elemanına da relevans skoru atamış olduk; bu skorlar filtre ağırlıklarının ne kadar kritik olduğuna dair bir ölçü verir. Bu  $L$  uzunluğundaki skorlar içinden **en büyüğü**, modelin  $i$  için  $j$  etkileşiminde en çok hangi gecikmeyi kullandığını (en önemli gördüğünü) gösterir <sup>80</sup> . Dolayısıyla  $i$  için  $j$  nedensel okunu grafiğe eklerken, üzerindeki gecikme değeri olarak  $\arg\max_{\tau} R_{K_{i,j}[\tau]}$  (en büyük skor alan gecikme  $\tau$ ) değeri atanır <sup>80</sup> . Bunu (20) nolu denklemle ifade etmişlerdir:

$$(20) \tau_{i \rightarrow j}^* = \arg\max_{\tau} R_{K_{i,j}[\tau]}$$

Elde edilen  $\tau_{i \rightarrow j}^*$ ,  $i$  serisinin  $j$  serisine etkisinin kaç birim zaman sonra gerçekleştiğini tahmin eder <sup>80</sup> .

Yukarıdaki adımlar her bir seri için uygulandığında, tüm serileri içeren yönlü ve gecikmeli bir nedensel ilişki grafi elde edilir. CausalFormer bu grafi çıktı olarak üretir. Kısacası, CausalFormer eğitilmiş

modelden **nedensel skorlar çıkarıp bunları filtreleyerek** (kümeler, eşikler yardımıyla) anlamlı bağlantılar haline getirir. Böylelikle başta tanımladığımız  $G=(V,E)$  grafi inşa edilmiş olur <sup>81</sup> .

## Deneyler ve Sonuçlar

Makalenin deneysel çalışmalarında CausalFormer, çeşitli yapay ve gerçek veri setleri üzerinde test edilmiştir. Sonuçları da diğer mevcut yöntemlerle karşılaştırılarak değerlendirilmiştir. Bu bölümde kullanılan veri setlerini, karşılaştırma yapılan yöntemleri, modelin ayarlarını ve elde edilen performansı özetleyeceğiz.

### 5.1 Veri Setleri

**Yapay (Sentezlenmiş) Veri Setleri:** Öncelikle, yer kontrolünün tamamen araştırmacıda olduğu **dört farklı sentetik zaman serisi verisi** oluşturulmuştur <sup>82</sup> . Bu veri setleri literatürde yaygın olarak incelenen temel nedensel grafik yapılarını temsil etmek üzere tasarlanmıştır: **“Elmas” yapısı**, **“Aracı (mediator)” yapısı**, **“V-şekli”** ve **“Çatal (fork)” yapısı** <sup>83</sup> . Örneğin, Elmas yapıda 4 zaman serisi bulunur; S1 serisi hem S2’yi hem S3’ü etkiler, bu ikisi de birleşerek S4’ü etkiler (nedensel grafiği bir elmas biçiminde olur). V-şekli yapıda iki bağımsız neden bir ortak sonucu etkiler (V harfi gibi). Her bir yapay veri setindeki seriler arası ilişkiler ve gecikmeler Şekil 7’de görselleştirilmiştir <sup>83</sup> . Bu veri setlerinde her bir zaman serisi belirli bir temel fonksiyon (lineer veya basit doğrusal olmayan) ile ebeveynlerinden üretilmiş ve üzerine ortalama 0, standart sapma 1 olacak şekilde Gauss (beyaz) gürültü eklenmiştir <sup>83</sup> . Elmas yapıdaki veri 4 seriden oluşurken, diğer üç yapı 3 seriliktir. Her serinin uzunluğu 1000 zaman adımı olarak üretilmiştir <sup>83</sup> .

**Simüle Edilmiş Lorenz-96 İklim Verisi:** Gerçekçi ama kontrol edilebilir bir senaryo olması açısından, popüler bir kaotik sistem modeli olan **Lorenz-96** kullanılarak bir veri seti oluşturulmuştur <sup>84</sup> . Lorenz-96, meteoroloji ve iklim dinamiklerinde kullanılan, her biri bir coğrafi bölgeyi temsil eden değişkenlerin birbiriyle etkileşimini tanımlayan bir sistemdir. Her değişkenin evrimi bir diferansiyel denklem ile verilir; komşu değişkenlerle etkileşim, kendine özgü kaotik davranışlar üretir. Makalede Lorenz-96 sistemi 10 değişken ile koşturulmuş ve 1000 zaman birimi boyunca simüle edilmiştir <sup>85</sup> . Denklem (21), Lorenz-96 modelinin formülünü özetlemektedir:

$$(21) \quad \frac{dx_i}{dt} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F,$$

bu denkleme göre her  $x_i$ , bir önceki iki ve bir sonraki bir komşusuyla etkileşir ve  $F$  parametresi sistemi süren dış kuvvettir <sup>86</sup> .  $F$  değerini artırdıkça sistem daha kaotik hale gelir. Bu simülasyonda  $F$  uygun bir değerde seçilmiştir (genelde 8.0 gibi bir değer Lorenz sistemini kaotik moda sokar) <sup>86</sup> . Lorenz-96 verisi, doğrusal olmayan ve yoğun etkileşimli nedensel yapıları içerdüğinden, modelin karmaşık durumlarla baş etme kapasitesini test eder.

**Gerçek fMRI (Beyin Sinyali) Verisi:** Makale, gerçek dünyadan bir nedensel keşif problemine örnek olarak **fonksiyonel manyetik rezonans görüntüleme (fMRI)** verilerini kullanmıştır <sup>87</sup> . fMRI verisi, beyindeki belirli bölgelerin zaman içindeki aktivasyon seviyelerini (BOLD sinyalleri) içerir. Bu sinyaller arasındaki nedensel ilişkiler, **beyin bölgelerinin fonksiyonel etkileşimlerini** anlamak açısından önemlidir. Kullanılan veri, 28 farklı beyin ağı için kaydedilen zaman serilerinden oluşmaktadır <sup>87</sup> . Her bir “ağ”, 5, 10, 15 veya 50 adet zaman serisi (beyin bölgesi) içerebilir, yani toplam değişken sayısı ağa göre değişir <sup>88</sup> . Zaman serilerinin uzunlukları da 50 ile 5000 zaman adımı arasında değişmektedir (deney süresine göre) <sup>87</sup> . Bu çeşitlilik, modelin hem az veri bulunan senaryolarda hem de çok veri bulunan senaryolarda test edilmesini sağlar. fMRI verisinin *gerçek nedensel grafi* elbette bilinmez (yerden

ölçüm ile doğrulanamaz) ancak sinirbilim alanında bazı **beklenen bağlantılar** literatürde mevcuttur; ayrıca burada model sonuçları diğer algoritmalarla karşılaştırılarak değerlendirilmiştir.

**Okyanus Suyut Sıcaklığı (SST) Verisi:** Yazarlar ayrıca modelin pratik faydasını göstermek amacıyla bir iklim verisi üzerinde uzun vadeli nedensel ilişkiler aramışlardır <sup>89</sup>. NOAA tarafından sağlanan **Kuzey Atlantik denizi yüzey sıcaklığı (Sea Surface Temperature, SST)** verisi, 4°x4° derecelik bir grid üzerinde 260 noktada ölçülen aylık sıcaklık değerlerini içerir <sup>90</sup> <sup>91</sup>. Bu veri 2013-2022 arasındaki 10 yıllık dönemi kapsar (her bir nokta için 97 zaman adımı) <sup>89</sup>. SST değişimleri, iklim sisteminde **okyanus akıntıları** gibi faktörlerle ilişkilidir. Bu veri seti üzerinde CausalFormer kullanılarak, belirli bölgelerin diğerlerini ısıtması veya soğutması yönündeki uzun vadeli ilişkiler incelenmiştir. (Bu özel çalışma sonuçları **Vaka İncelemesi** bölümünde ayrıntılı ele alınacaktır.)

## 5.2 Karşılaştırılan Yöntemler (Baseline Modeller)

CausalFormer'ın başarısını değerlendirmek için, yazarlar onu literatürdeki çeşitli derin öğrenme tabanlı nedensel keşif yöntemleriyle karşılaştırmıştır. Seçilen temel (baseline) yöntemler ve kısa açıklamaları şu şekildedir:

- **cMLP & cLSTM (2021):** Tank ve arkadaşlarının çalışmaları olan cMLP (causal MLP) ve cLSTM, çok katmanlı perceptron ve LSTM ağlarını **Granger nedenselliği** prensibiyle kullanan yöntemlerdir <sup>14</sup>. Bu modeller, her bir aday nedensel ilişki için bir nöron bağlantısı tahsis ederek, eğitim sonunda hangi bağlantıların aktif (büyük) kaldığına bakar. Yani belirli girdilerdeki aktif ağırlıklar **nedensel ilişki** olarak yorumlanır. cLSTM de benzer şekilde, LSTM ağının kapı mekanizmalarındaki etkileşimleri inceler.
- **TCDF (2019):** Nauta ve arkadaşlarının geliştirdiği Temporal Causal Discovery Framework, konvolüsyonel sinir ağı ile dikkat mekanizmasını bir arada kullanarak zaman serilerinde nedensellik bulur <sup>92</sup>. Model her bir hedef seri için küçük bir CNN+attention ağı eğitir ve sonrasında o ağı dikkat ağırlıkları ile konvolüsyon filtrelerine bakarak nedensel ilişkileri çıkarır <sup>12</sup>. Yukarıda da tartıştığımız gibi, TCDF **lokal parametre analizi** yapan bir yöntemdir ve kısmen eksik haritalama yapıyor olabilir; ama dönemi için popüler bir temel yöntemdir.
- **DVGNN (2023):** “Dynamic Variational Graph Neural Network” adlı bu yöntem, zaman serileri için **graf tabanlı** bir nedensellik öğrenme yaklaşımıdır <sup>93</sup>. DVGNN, zaman serilerini bir grafik üzerinde düğümler kabul edip Graph Convolutional Network (GCN) kullanarak temsil öğrenir. Bununla birlikte, olası graf yapıları üzerinden de bir **varyasyonel difüzyon** süreci işletir. Sonuçta model, bir istatistiksel çıkarım yöntemi gibi graf yapısını optimizasyonla belirler.
- **CUTS (2023):** “Causal discovery from Irregular Time Series” adlı bu yöntem, özellikle **düzensiz aralıklı** (irregular) zaman serilerindeki nedenselliği keşfetmek için geliştirilmiştir <sup>94</sup>. CUTS önce eksik gözlemleri olan veya farklı zaman ölçeklerinde örneklenen verileri, **Delayed Supervision GNN (DSGNN)** adı verilen bir modelle uygun şekilde imputasyon yaparak düzenli hale getirir; ardından bir grafik üzerindeki nedenselliği öğrenir. Model, grafik öğreniminde de seyreklik cezaları içermektedir. Bizim problemimizde veriler düzenli örneklenmiş olsa da, CUTS genel performans kıyaslaması için dahil edilmiştir.

Bu yöntemler dışında Granger'ın klasik VAR analizi veya PC algoritması gibi istatistiksel yöntemlerle de kıyaslama yapılabilir, ancak yazarlar odak noktası olarak **derin öğrenme tabanlı** yaklaşımları seçmişlerdir. Zira CausalFormer bu kategori içerisinde yenilik getirmektedir.

### 5.3 Deneysel Kurulum

**Model Eğitim Ayarları:** CausalFormer modeli, her veri seti için özel olarak ayarlanmıştır. Zira veri setlerinin boyutları ve karmaşıklıkları farklı olduğu için, evrensel bir hiper-parametre grubu yerine **duruma özel optimizasyon** tercih edilmiştir <sup>95</sup>. Örneğin, sentetik veri setleri yalnızca 3-4 zaman serisi içerdiğinden, modelin parametre sayısını küçük tutmak ve aşırı öğrenmeyi önlemek için daha az sayıda attention başlığı (\$H\$), daha düşük boyutta gömme vektörleri (\$d\$) kullanılmıştır <sup>95</sup>. Buna karşılık Lorenz-96 gibi hem değişken sayısı hem de ilişkilerin karmaşıklığı daha yüksek olan verilerde, model boyutları biraz büyütülmüş ve daha fazla konvolüsyon çekirdeği gibi özellikler eklenmiştir <sup>96</sup>. fMRI veri setinde ise, özellikle **normalizasyon (seyreklik cezası) terimleri** kaldırılmıştır (\$\gamma = \zeta = 0\$) alınmış), çünkü beyin ağlarındaki gerçek nedensel bağlantıların sayısı diğerlerine göre fazla olabilir; bunları budamak adına ceza uygulanmamıştır <sup>97</sup>. Tüm modeller He initialization ile ağırlık başlatması yapılmış ve öğrenme süreci **Adam** optimizasyon algoritması ile yönetilmiştir <sup>98</sup>. Eğitim sırasında bir **erken durdurma (early stopping)** stratejisi kullanılmıştır, yani doğrulama hatası belirli bir süre iyileşmezse eğitim sonlandırılmıştır <sup>98</sup>.

**Değerlendirme Metrikleri:** İki temel değerlendirme ölçütü kullanılmıştır: (1) **Nedensel ilişki keşfinde doğruluk**, (2) **Gecikme (lag) keşfinde doğruluk**. İlk metrik için genellikle **Precision (Kesinlik)**, **Recall (Duyarlılık)** ve **F1-Skoru** hesaplanır. Precision, modelin bulduğu nedensel ilişkilerin ne kadarının gerçek olduğu; Recall ise gerçek nedensel ilişkilerin ne kadarını modelin yakalayabildiği oranıdır. F1-Skoru bu ikisinin harmonik ortalamasıdır. Gerçek verilerde gerçek nedensel graf bilinmediği için bu metrikler ancak sentetik ve Lorenz verisi gibi kontrollü senaryolarda hesaplanabilir. fMRI için yazarlar "beklenen bağlantılar" üzerinden benzer bir değerlendirme yapmıştır. İkinci metrik olan **Gecikme Hassasiyeti (Precision of Delay, PoD)**, modelin tespit ettiği nedensel ilişkinin gecikme değeriyle gerçek gecikme değeri arasındaki farkı ölçer. Bu da gecikme tahmininin isabet oranını gösterir. PoD yüksek ise model, nedensel etkinin kaç adım sonra geldiğini doğru bulmuş demektir.

**Adil Karşılaştırma:** Her bir temel yöntemin kendi makalesinde önerilen ayarları kullanılmıştır. Ayrıca, DVGNN ve CUTS gibi yöntemler doğrudan bir nedensel graf vermez, her bir ilişki için bir skor matrisi üretir. Bu yöntemlerden çıkan skorları nedensel ilişkiye çevirmek için, CausalFormer'da olduğu gibi \$k\$-ortalama kümeleme uygulanmış ve en yüksek kümeler seçilmiştir <sup>99</sup>. Bu sayede, değerlendirmede tüm modeller için benzer kriterler uygulanmış olur. TCDF ve cMLP gibi yöntemler ise zaten içsel olarak bir eşikleme veya önem sıralaması ile sonuç verirler.

### 5.4 Sonuçlar ve Tartışma

**Genel Nedensel Keşif Performansı:** Tablo 1'de, CausalFormer ile karşılaştırma yöntemlerinin çeşitli veri setlerindeki F1-skorları verilmektedir <sup>100</sup>. Sonuçlar oldukça dikkat çekicidir:

- Sentetik veri setlerinde (Elmas, Aracı, V-şekli, Çatal), CausalFormer hemen her durumda en yüksek veya en yükseklerden biri F1-skoruna ulaşmıştır <sup>100</sup>. Özellikle V-şekli ve Çatal yapılarında fark belirgindir; CausalFormer bu karmaşık ilişkileri yüksek doğrulukla çıkarabilirken, bazı yöntemler (örn. DVGNN, CUTS) düşük kalmıştır. Bu durum, önerilen modelin **farklı temel nedensel yapıları doğru tanıyabildiğini** gösteriyor <sup>100</sup>. Zira her sentetik yapı, modelin farklı bir özelliğini sınar: Örneğin V-şekli yapı, ortak bir sonucun iki nedeni ayrıştırılabilir mi test eder; Elmas yapı, dolaylı yollu etkileri (\$S\_1 \rightarrow S\_4\$ direk yok ama \$S\_2, S\_3\$ üzerinden dolaylı) yakalıyor mu bakar. CausalFormer bu tip karmaşıklıkları başarıyla yönetmiştir.
- Lorenz-96 simülasyonunda, CausalFormer yine en yüksek F1-skoruna ulaşmıştır (0.69 civarında) ve diğer tüm yöntemleri geride bırakmıştır <sup>101</sup>. Lorenz verisi non-lineer ve oldukça bağlı bir yapı içerdiğinden, klasik yöntemler bir hayli zorlanmıştır (bazıları 0.46-0.58 F1'de kalmış).

CausalFormer ise derin modelin temsil gücünü tam kullanabildiği ve bunu yorumlama tekniğiyle birleşik ele aldığı için **en iyi sonucu** vermiştir <sup>101</sup> . Bu, modelin derin öğrenmenin güçlü temsil kabiliyetini nedensel keşifte avantaja çevirebildiğinin kanıtıdır <sup>101</sup> .

- Gerçek fMRI verisinde, net bir "doğru grafik" olmamasına rağmen diğer yöntemlerle karşılaştırma yapıldığında yine CausalFormer'ın en yüksek ortalama F1 değerini sağladığı görülür ( $0.66 \pm 0.09$ ) <sup>102</sup> <sup>100</sup> . Özellikle dikkat çeken nokta, CausalFormer'ın **tutarlı bir şekilde** precision ve recall arasında dengeli kalmasıdır; bazı yöntemler yüksek precision ama düşük recall veya tam tersi sonuçlar verebilirken, CausalFormer her ikisinde de makul değerler olarak dengeli bir F1 yakalamıştır. Bu da yöntemimizin **hem yanlış pozitifleri kontrol etmede hem de gerçek pozitifleri yakalamada** başarılı olduğunu gösterir <sup>100</sup> .

Genel olarak, CausalFormer tüm deneylerde en iyi veya rakiplerine eşdeğer performans göstererek **state-of-the-art** düzeyde olduğunu kanıtlamıştır <sup>103</sup> <sup>104</sup> . Bu başarının altındaki neden, modelin derin öğrenmenin temsil gücünü tam anlamıyla kullanırken, yorumlanabilirlik ile onu sınırlandırıp doğru yöne kanalize etmesidir. Yani CausalFormer, saf bir kara-kutu gibi davranmadığından, gereksiz ilişkileri öğrenmeye meyilli değildir; RRP gibi bir mekanizma ile gerçekten **anlamli ilişkilere odaklanır**.

**Gecikme (Lag) Tahmin Performansı:** Bir nedensel keşif modelinin sadece "kim kime etki ediyor"u değil, aynı zamanda "ne kadar süre sonra etki ediyor"u da doğru bulması istenir. CausalFormer, içerisinde barındırdığı çoklu-çekirdekli konvolüsyon yapısı sayesinde, her ilişki için bir gecikme değeri de tahmin edebilmektedir (en büyük katkı gelen gecikme  $\tau$  olarak) <sup>105</sup> . Rakip modellerden cMLP kısmen gecikme yakalayabilir (çünkü bağlantı ağırlıkları gecikmeye göre ayrı ayrı mevcuttur) ve TCDF ise dilate konvolüsyon kullandığı için gecikme tespiti yapabilir; ancak diğerleri gecikme bilgisi sağlamaz. Tablo 2'de, sadece bu üç yöntem (cMLP, TCDF, CausalFormer) için **Precision of Delay (PoD)** değerleri kıyaslanmıştır <sup>105</sup> <sup>106</sup> . İlginç biçimde, gecikme tahmininde **CausalFormer en yüksek değerleri alamamıştır** <sup>107</sup> . Örneğin Elmas ve Mediator verilerinde TCDF %92 ve %97 PoD sağlarken, CausalFormer %74 ve %63'te kalmıştır <sup>108</sup> . Özellikle V-şekli ve Fork yapılarında TCDF %100 doğru lag bulurken, CausalFormer yarıdan az isabet sağlamıştır (tabloda %59 ve %46) <sup>109</sup> . Bu sonuç ilk bakışta modelimizin bir eksiği gibi görünse de, yazarlar bunun nedenini açıklamışlardır:

TCDF modeli, her bir ilişki için girdi verisini farklı dilasyon değerleriyle konvolüle ederek **en uygun gecikmeyi yapısal olarak aramaktadır**; cMLP de benzer biçimde yakın zaman adımlarına daha ağır ceza koyarak modelini, mecburen belli gecikmeleri seçmeye iter <sup>110</sup> . CausalFormer ise, tahmin görevi sırasında **tüm pencere boyunca veriyi adil biçimde kullanır** (ne çok yakın geçmişe ne çok uzak geçmişe özel ceza vermez) <sup>111</sup> . Bu nedenle, bazı durumlarda bir etkinin tam hangi anda doruğa çıktığını değil, kümülatif etkisini öğrenmiş olabilir. Kısaca, CausalFormer **nedensel ilişkiyi doğru bulsa bile gecikmeyi tam isabetle bulamayabilir**. Yazarlar bunu bir gelecek çalışmaya açık alan olarak belirtmiştir: Konvolüsyon sürecine ekstra kısıt veya ceza terimleri ekleyerek, özellikle gecikme tespitini iyileştirme potansiyeli vardır <sup>112</sup> . Böyle bir geliştirme yapılırken, tabi ki genel nedensellik tespit performansının korunması gözetilecektir (fazladan ceza eklerken ana F1 skorunun düşmemesi istenir) <sup>113</sup> .

**Not:** fMRI veri setinde gerçek nedensel gecikmeler bilinmediği için orada PoD raporlanmamıştır <sup>114</sup> .

**Çıkarım:** CausalFormer, "hangi ilişkiler var" sorusuna çok başarılı yanıt verirken, "bu ilişkinin gecikmesi tam olarak nedir" sorusunda mevcut haliyle rakiplerinden biraz geride kalabilmektedir. Ancak bu, modele eklenecek ek düzenlemelerle geliştirilebilecek bir yön olarak görülmektedir.

## 5.5 Ablasyon Analizleri

Modelimizin başarısını sağlayan bileşenlerin önemini doğrulamak için, yazarlar bir dizi **ablasyon (bileşen çıkarma) deneyi** yapmışlardır. Bu deneylerde CausalFormer'dan belirli bir bileşen çıkarıldığında veya değiştirdiğinde performansın nasıl etkilendiğine bakılmıştır. Ablasyon çalışmaları, en karmaşık gerçek veri seti olan **fMRI** üzerinde gerçekleştirilmiştir (28 ağın ortalaması alınarak) <sup>115</sup>. Karşılaştırılan varyantlar şunlardır <sup>116</sup>:

- 1. Yorumlama olmadan (w/o interpretation):** Bu versiyonda, model eğitildikten sonra RRP gibi herhangi bir ayrıştırma yapılmaz. Nedensel skorlar, **doğrudan dikkat matrisi \$A\$ ile konvolüsyon ağırlıklarının değerlerinden** alınır. Yani bu, CausalFormer'ın getirdiği yorumlama yeniliğini kullanmayan bir yaklaşımı temsil eder. Temelde TCDF tarzı bir çıkarım diyebiliriz (dikkat ağırlığı büyükse ilişki var varsay gibi).
- 2. Relevans olmadan (w/o relevance):** Bu varyantta RRP'nin relevans hesaplama kısmı çıkarılmış, sadece **gradyan büyüklükleri** nedensel skor olarak kullanılmıştır. Yani her potansiyel  $i$  için  $\frac{\partial \text{loss}}{\partial x_i}$  gibi bir değer hesaplanıp ilişki sıralaması yapılmıştır. Bu, saf gradient bazlı bir açıklama yöntemine denk düşer (LRP yerine).
- 3. Gradyan olmadan (w/o gradient):** Bu ise tam tersine, RRP ile relevans skorları hesaplanıp **gradyan modülasyonu yapılmamış** halidir. Yani sadece  $R_i$  skorları kullanılmıştır,  $C_i = R_i$  kabul edilmiştir. Bu da saf LRP-türevi bir yaklaşımı yansıtır.
- 4. Bias olmadan (w/o bias):** RRP yönteminde bias için ayrı relevans hesabı yapmayı devre dışı bırakan versiyondur. Yani (15)-(16) denklemindeki bias paylaşımını yerine, bias tamamen yok sayılarak  $R_i = \frac{x_i}{\sum_k x_k} R_{\text{out}}$  formülü kullanılmıştır. Böylece bias'ın bir kısmını yutması önlenmiş olur, ancak belki yanlış yerlere fazla krediyi de iade etmiş olur.
- 5. Çoklu konvolüsyon çekirdekleri olmadan (w/o multi conv kernel):** Bu varyantta CausalFormer'daki konvolüsyon bloğu sadece **tek bir çekirdek** kullanır (yani her  $i$  için birden fazla bağımsız çekirdek yoktur, klasik bir tek-kernel konvolüsyon yapılıır). Böylece model, her ilişki için paralel veri akışları kuramaz, tüm etkiyi tek bir filtre ile temsil eder.

Tablo 3'te bu varyantların fMRI veri setindeki ortalama **Precision, Recall, F1** değerleri listelenmiştir <sup>117</sup>. Orijinal CausalFormer'ın değerleri en altta verilmiş (Precision ~0.80, Recall ~0.59, F1 ~0.66) <sup>118</sup>. Çıkarılan bileşenlerin sonuçlarına baktığımızda:

- **Yorumlama olmadan:** Precision ~0.47, Recall ~0.45, F1 ~0.44 gibi oldukça düşük bir performans görüyoruz <sup>119</sup>. Bu, beklenen bir sonuçtur; zira modelin sadece dikkat ağırlıkları ve konvolüsyon ağırlıklarına bakılması, özellikle dolaylı ilişkilerde hata yapmasına yol açar. Bu durumda model pek çok ilişkiyi atlamış (Recall düşmüş) ve yanlış eşleşmeler bulmuş (Precision düşmüş) görünüyor. **En büyük performans kaybı**, yorumlama bileşeninin tamamen kaldırılmasında yaşanmıştır <sup>120</sup>. Bu, RRP tabanlı nedensellik dedektörünün modelin başarısındaki en kritik unsur olduğunu doğrular: Bütün Transformer'ı yorumlamak yerine sadece lokal parametrelere bakmak, temporal nedensel keşifte ciddi bir eksiklik yaratmaktadır <sup>120</sup>.
- **Relevans olmadan (sadece gradyan):** Precision ~0.64, Recall ~0.44, F1 ~0.50 elde edilmiştir <sup>119</sup>. Bu versiyon, aslında model çıktısına hangi girdilerin en hassas olduğunu bulur (gradyanı büyük olanları seçer). Sonuçlara göre Precision fena değildir (0.64) ama Recall oldukça düşüktür (0.44).



Yani sadece gradyana bakmak, bazı gerçek nedensel ilişkileri kaçırmamıza neden olmuştur (model belki bazı nedenleri düşük gradyanlı diye almamış, ancak onların net etkisi önemliydi). **Relevans skoru** ise daha bütünsel bir katkı ölçüsü olduğu için recall'u yükseltiyor; gradyan ise **kesinlik** sağlıyor denilebilir <sup>121</sup>. Nitekim orijinal modele kıyasla, sadece gradyanda recall epey düştü, precision nispeten iyi kaldı. Bu, **RRP'nin (relevance) özellikle recall'ü artırdığını**, yani daha fazla gerçek ilişkiyi ortaya çıkardığını gösterir <sup>121</sup>.

- **Gradyan olmadan (sadece relevans):** Precision ~0.60, Recall ~0.54, F1 ~0.54 olarak raporlanmıştır <sup>119</sup>. Bu durumda recall, sadece gradyan versiyonuna göre daha yüksek (0.54 vs 0.44), precision ise biraz daha düşük (0.60 vs 0.64). Yine beklendiği gibi: Relevans yöntemi potansiyel tüm ilişkileri yakalamaya çalışır (false negative az, yani recall iyi), fakat yanlış pozitif de olabilir (false positive artabilir, precision düşer). Yazarların ifadesiyle, **"relevance decomposition recall'u artırmaya odaklanırken, gradyanlar kesinliği artırmaya daha yatkın"**dır <sup>121</sup>. Bu nedenle orijinal modelde her ikisinin kombinasyonu (grad modülasyonu) en iyi dengeyi sağlar.

- **Bias olmadan:** Precision ~0.79, Recall ~0.44, F1 ~0.55 elde edilmiştir <sup>119</sup>. Bias çıkarıldığında precision neredeyse orijinal kadar yüksek kalmış, fakat recall bariz düşmüştür (0.59'dan 0.44'e). Bu demektir ki, bias'ı hesaba katmayan yöntem **bazı ilişkileri tamamen atlamıştır**. Bias'ı yok saydığımızda, modelin bias terimleri tarafından açıklanan kısımları muhtemelen yanlış yere girdi etkisi olarak dağıtmış oluyoruz veya tam tersi bazı girdilerin etkisini ısıkılıyor. Bias'ı ayrıca ele alan orijinal RRP, modelin sırf bias yüzünden kaçırabileceği ilişkileri de yakalamasına yardımcı oluyor. Sonuçta bias'lı RRP'nin recall'u iyileştirdiğini görüyoruz <sup>122</sup>. Yazarlar, bias teriminin ayrıştırılmasının özellikle **yanlış biçimde girdilere atfedilebilecek katkıları önlediğini** ve böylece recall'u artırdığını belirtiyor <sup>123</sup>.

- **Çoklu çekirdek olmadan:** Precision ~0.74, Recall ~0.56, F1 ~0.61 bulunmuştur <sup>124</sup>. Çoklu çekirdek yerine tek çekirdek kullanan model, orijinaline yakın ama her metrikte biraz daha düşük kalmıştır (özellikle F1 0.66'dan 0.61'e düşmüş). Bu, beklenen bir sonuçtur: Çoklu çekirdek kullanımı her bir seri çiftine ilişkin bağımsız bilgi akışları sağlayarak modelin hem kesinliğini hem duyarlılığını bir miktar artırmıştı <sup>125</sup>. Tek çekirdekli modelde belki bazı ilişkiler **"genelleştirilmiş"** şekilde temsilden kaçmış olabilir (precision düşebilir) veya tek kanala sıkıştığı için bazı zayıf etkiler baskılanmış olabilir (recall düşebilir). Görüldüğü üzere her iki ölçüt de bir miktar gerilemiş. Bu da **çoklu-çekirdekli konvolüsyonun** CausalFormer'a kayda değer bir katkı sağladığını gösterir <sup>125</sup>.

Bu ablation çalışmalarından çıkan genel mesaj, CausalFormer'ın tasarımındaki her bir yeniliğin performansa olumlu katkısı olduğudur. Özellikle **RRP ile modelin yorumlanması** en kritik bileşen olarak öne çıkmıştır (olmazsa olmaz) <sup>120</sup>. Gradyan modülasyonunun ve bias ayrıştırmasının da hatırı sayılır etkileri olduğu, bunların ihmal edilmesi durumunda performansın bariz düştüğü görülmüştür <sup>121</sup>. Çoklu konvolüsyon çekirdekleri de modelin doğruluk ve duyarlılığına katkıda bulunmaktadır <sup>125</sup>. Bu sonuçlar, makalenin önerdiği her bileşenin gerekçesini deneysel olarak doğrulamaktadır.

**Tablo 3 Özeti:** (Açıklayıcı olarak ekleyelim) Orijinal CausalFormer F1  $\approx$  0.66 iken, yorumlama yoksa ~0.44, relevans yoksa ~0.50, gradyan yoksa ~0.54, bias yoksa ~0.55, multi-kernel yoksa ~0.61 F1 elde edilmiş <sup>117</sup> <sup>126</sup>. En fazla etki yorumlama (detektör) kısmında; sonrasında relevans+grad birlikte kullanımı (ikisi de önemli), bias da bir miktar etki ediyor; multi-kernel ise en az fark ettiren ama yine de pozitif etkili parça.

## 5.6 Vaka İncelemeleri ve Uygulamalar

Bu bölümde, CausalFormer'ın pratikte ortaya koyduğu nedensel grafikleri yorumlamak üzere makalede sunulan iki örneği ele alacağız: Birincisi fMRI verisinden seçilen bir ağ için nedensel grafik, ikincisi ise Kuzey Atlantik SST verisi için bulunan uzun vadeli nedensel ilişkiler. Bu örnekler, modelin çıktılarının **anlaşılabilirliğini ve gerçek dünyadaki geçerliliğini** göstermesi bakımından önemli.

**fMRI Beyin Ağı Örneği:** Makalede **fMRI-15** olarak anılan, 15 zaman serisinden oluşan bir beyin ağı için, farklı modellerin çıkardığı nedensel grafiğin görsel karşılaştırması verilmiştir (Şekil 8) <sup>127</sup>. Bu görselde siyah oklar gerçek pozitif (hem gerçek bağlantı var hem model bulmuş), kırmızı oklar yanlış pozitif (model bulmuş ama gerçekte yok) ve kesikli oklar da modelin kaçırdığı gerçek bağlantıları (false negative) göstermektedir <sup>128</sup>. CausalFormer'ın grafiğinde, yalnızca **iki tane** kırmızı ok (yani gerçekte olmayan ilişkiyi var sanmış) ve **bir tane** kesikli ok (yani bir gerçek ilişkiyi atlamış) görülmektedir <sup>129</sup>. Diğer yöntemlere baktığımızda (cMLP, TCDF, CUTS, DVGNN), yanlış pozitif ok sayısının daha fazla olduğu ve hatta bazı durumlarda **nedensel yönün tersine döndüğü** hatalar yaptığı belirtilmiştir <sup>130</sup>. Örneğin, cMLP, TCDF ve CUTS modelleri bir durumda X serisinin Y'yi etkilediğini iddia ederken, gerçekte **tam tersi** (Y'nin X'i etkilediği) biliniyormuş; CausalFormer ise doğru yönü saptamıştır <sup>131</sup>. Bu, modelimizin sadece bağlantı var-yok konusunda değil, yön tayini konusunda da başarılı olduğunu gösterir. Yön hatası genelde Granger testlerinde sık görülür (iki yönlü korelasyon varsa yanlış yöne yordama riski olur), ancak CausalFormer dolaylı etkileri de hesaba kattığı için bu tuzağa daha az düşer. Sonuç olarak, fMRI örneğinde CausalFormer **en az hatayı yapan** model olarak öne çıkmıştır. Bu durum, beyin ağı gibi karmaşık ve gürültülü bir ortamda bile modelin görece güvenilir ilişkiler yakalayabildiğine işaret eder <sup>130</sup>.

**Şekil:** Kuzey Atlantik'teki deniz yüzeyi sıcaklığı (SST) zaman serileri arasında CausalFormer tarafından keşfedilen nedensel ilişkiler. Bu ilişkiler, bölgedeki **okyanus akıntılarının yönü ve etkisiyle uyumlu** görünmektedir. Turuncu oklar, **güneyden kuzeye** (S→N) yönünde nedenselliği belirtir, yani daha düşük enlemlerdeki sıcak suların daha yüksek enlemlerdeki bölgelerin SST değişimine yol açtığını (sıcak akıntıların ısı taşıması) gösterir. Mor oklar ise **kuzeyden güneye** (N→S) nedenselliği gösterir, yani yüksek enlemlerdeki soğuk suların daha güneydeki bölgelerin yüzey sıcaklığını etkilediği durumları temsil eder. CausalFormer'ın ortaya çıkardığı bu nedensel oklar, **Kuzey Atlantik'teki ana okyanus akıntılarının dağılımıyla büyük ölçüde tutarlıdır** <sup>132</sup>. Örneğin, Gulf Stream ve onun devamı olan North Atlantic Drift boyunca birçok S→N yönlü nedensel bağlantı bulunmuştur; bu da düşük enlemlerle sıcak okyanus akıntılarının, kuzeydeki daha soğuk deniz bölgelerini ısıttığı anlamına gelir <sup>132</sup>. Benzer şekilde, Grönland çevresinde tespit edilen N→S yönlü nedensel oklar, bu bölgedeki soğuk akıntılarının (ör. Labrador Akıntısı gibi) güneydeki deniz yüzeyi sıcaklıklarını düşürücü etkisini göstermektedir <sup>132</sup>.

Bunun yanı sıra, CausalFormer sonucu elde edilen grafikte **Batı Atlantik** kısmında doğu tarafına kıyasla çok daha **karmaşık ve yoğun** nedensel bağlantılar görülmektedir. Bu durum, fiziksel gerçeklikle de örtüşmektedir: Batı Atlantik'te (özellikle Gulf Stream'in çıkış bölgesinde) sıcak ve soğuk akıntılarının, ayrıca atmosferik etkileşimlerin oluşturduğu dalgalanmalar daha fazladır. Doğu Atlantik'te ise Kanarya Akıntısı gibi daha **yavaş hareket eden** (dolayısıyla etkisi daha lokal kalan) akıntılar bulunduğu için, nedensel etkileşimler görece seyrek olur <sup>132</sup> <sup>133</sup>. CausalFormer grafiğinde de batı tarafta çok sayıda çapraz bağlantı varken, doğu tarafta daha az ve belirgin yönlerde bağlantılar saptanmıştır. Genel olarak, SST verisi üzerindeki bu inceleme, modelimizin **coğrafi ve fiziksel olarak anlamlı nedensel ilişkiler yakalayabildiğini** göstermiştir. Kuzey Atlantik'teki okyanus akıntılarının güney-kuzey doğrultusunda ısı taşıdığı ve kuzey-güney doğrultusunda soğuk su gönderdiği bilinir; CausalFormer da bulgularıyla bunu doğrular niteliktedir <sup>132</sup> <sup>134</sup>. Yazarlar bu sonuçların, modelin iklim bilimi gibi alanlarda yararlı içgörüler sağlayabileceğine işaret ettiğini vurgulamışlardır. Nitekim, uzun vadeli SST nedenselliğinin anlaşılması, **hava tahminleri, iklim modellemeleri ve deniz ekosistemi yönetimi** gibi uygulamalarda değerlidir <sup>135</sup>.

Son olarak, SST verisi örneğinde model parametrelerinin nasıl seçildiğine değinmekte fayda var. Yazarlar, Kuzey Atlantik'te tipik yüzey akıntı hızının  $\sim 0.3$  m/s olduğunu ve bu akıntıların bir 4x4 derecelik grid hücresinden komşu hücreye yaklaşık 1000 km mesafe kat etmesinin yaklaşık **38 gün** sürdüğünü hesaplamışlardır <sup>136</sup>. Bu nedenle SST verisini işlerken, bir zaman adımını 38 güne eşitleyen bir ölçeklendirme yapılmıştır <sup>136</sup>. Bu sayede, modeldeki 1 gecikme birimi  $\approx 1000$  km'lik coğrafi etkiye karşılık gelmiştir. Böyle bir ayar, modelin gerçekten **fiziksel süreçlere uygun** bir zaman ölçeğinde çalışmasını sağlamıştır. Örneğin sıcak bir su kütesinin kuzeye ilerleyip bir bölgeyi etkilemesi yaklaşık 3-4 adım (aylar) sürüyorsa, model de 3-4 lag'lı bir S $\rightarrow$ N ilişki bulmaktadır. Bu da modelin bulgularını yorumlamayı kolaylaştırmıştır. Yazarlar, CausalFormer çıktılarının genel olarak Kuzey Atlantik akıntı sistemleri ile **tutarlılık gösterdiğini ve modelin etkinliğini doğruladığını** belirterek vaka incelemesini sonlandırmışlardır <sup>134</sup>.

## 6 Sonuç ve Değerlendirme

Bu makalede, zaman serilerinde nedensel ilişkileri keşfetmek için yeni ve güçlü bir yaklaşım olan **CausalFormer** ayrıntılı şekilde tanıtılmıştır. CausalFormer, modern derin öğrenme mimarilerinden Transformer'ı, nedenselliğe uygun özel bileşenlerle donatarak ve bir yorumlanabilirlik mekanizmasıyla destekleyerek, mevcut yöntemlerin ötesine geçen bir performans sergilemiştir. Çalışmanın öne çıkan katkıları ve bulguları şöyle özetlenebilir:

- **Yorumlanabilir Transformer modeli:** Makale, nedensel keşif problemi için özel olarak yapılandırılmış bir transformer modeli sunmuştur. Bu model, içerdiği **çoklu-çekirdekli konvolüsyon katmanı** sayesinde her bir potansiyel nedensel etkiyi ayrı bir filtre olarak işler ve **zamansal öncelik kuralına** uygun şekilde veriyi kullanır <sup>137</sup>. Ayrıca **çok-değişkenli dikkat mekanizması**, seriler arasında karmaşık ilişki örüntülerini yakalamak üzere tasarlanmıştır. Bu tasarım, klasik yaklaşımların çoğunun göz ardı ettiği **eşzamanlı ve öz-nedensel** etkileşimleri dahi modelleyebilmektedir <sup>22</sup>.
- **Regresyon için katman bazlı ayrıştırma tekniği:** CausalFormer, eğitilmiş modelin tamamını açıklamak için **Regression Relevance Propagation (RRP)** yöntemini geliştirmiştir. Bu yöntem, sınıflandırma ağlarındaki LRP tekniğini regresyon ağlarına genelleyerek, her bir giriş serisinin çıktı üzerindeki katkısını katman katman hesaplar <sup>137</sup>. RRP ile birlikte sunulan yenilikler (bias ayrıştırması, gradyan modülasyonu vb.), Transformer gibi karmaşık bir modelin **küresel düzeyde yorumlanmasını** mümkün kılmıştır. Sonuçta modelin içinden elde edilen **nedensel skorlar**, tam bir nedensel graf inşa etmeye olanak vermiştir <sup>137</sup>.
- **Üstün deneysel performans:** Gerek sentetik testlerde, gerekse gerçek dünya verilerinde CausalFormer rakip yöntemleri belirgin şekilde geride bırakmıştır. Özellikle farklı yapısal motiflere sahip sentetik verilerde tümünü doğruya yakın saptayabilmiş; kaotik Lorenz sisteminde ve nörolojik fMRI verilerinde en yüksek nedensel keşif skorlarını elde etmiştir <sup>100</sup> <sup>103</sup>. Bu sonuçlar, derin öğrenme modellerinin temsil gücünün tamamını kullanıp bunu yorumlama ile desteklemenin **etkili bir strateji** olduğunu göstermektedir. CausalFormer, kara-kutu modelinin gücü ile istatistiksel yöntemlerin şeffaflığını başarılı bir biçimde bir araya getirmiştir.
- **Ablasyon ve analizlerle doğrulanan tasarım:** Modelin kritik bileşenlerinin her birinin (multi-kernel conv, RRP, gradient modülasyonu, bias ayrıştırması) performansla olumlu katkısı deneylerle gösterilmiştir <sup>126</sup>. Hiçbir bileşen gereksiz değildir; aksine çıkarıldığında belirgin bozulmalara yol açmaktadır. Bu da tasarımın bütünselliğini ve her parçasının amaca hizmet ettiğini kanıtlar niteliktedir.

- **Pratik uygulama potansiyeli:** Yapılan vaka incelemeleri, CausalFormer'ın sadece sayısal skorlar üretmekle kalmayıp, **yorumlanabilir ve anlamlı sonuçlar** verdiğini ortaya koymuştur. Beyin sinyali verisinde model, beklenen bağlantıları bulup ters bağlantı gibi hataları minimize edebilmiştir; iklim verisinde ise okyanus bilimine uygun nedensel desenler yakalamıştır. Bu durum, yöntemin bilimsel keşiflerde **hipotez üretici** bir araç olabileceğini düşündürmektedir. Araştırmacılar, model çıktısını inceleyerek yeni nedensel ilişkiler fark edebilir veya mevcut inançları teyit edebilirler. Örneğin iklim uygulamasında, SST sonuçları meteorologlara okyanus-atmosfer etkileşimlerini daha iyi anlama yönünde ipuçları verebilir <sup>132 134</sup>.

Özetle, CausalFormer çalışması **derin öğrenme ile nedensel keşif** arasında önemli bir köprü kurmaktadır. Geleneksel yöntemlerin kısıtlarını aşmak için derin modelleri kullanırken, bu modellerin yorumlanabilirliğini de sağlayarak sonuçları güvenilir kılmaktadır. Makale, gelecekte bu yaklaşımın çeşitli geliştirmelerle daha da iyileştirilebileceğine değinmiştir. Özellikle **gecikme tespiti** konusunda daha hassas olabilmek için konvolüsyon katmanına ek kısıtlar veya cezalar eklemek bir araştırma konusu olarak belirtilmiştir <sup>112</sup>. Ayrıca, modelin büyük ölçekte uygulanması (örneğin onlarca- yüzlerce değişkenli sistemlerde) ve olası hesaplama optimizasyonları da gelecekte ele alınabilir.

Sonuç itibarıyla, CausalFormer alan yazına yenilikçi bir katkı sunmuş ve **zamansal nedensellik keşfi problemini** çözmede yeni bir standart belirleme potansiyeli göstermiştir. Kodlarının açık kaynak olarak paylaşılması sayesinde <sup>138</sup>, araştırmacılar modeli farklı veri kümelerinde deneyebilir, geliştirebilir ve gerçek dünyadaki karmaşık nedensel ağları aydınlatmak için kullanabilirler. Bu çalışma, AI sistemlerinin sadece tahmin yapmakla kalmayıp **neden-sonuç ilişkilerini anlamlandırma** yönünde de ilerleyebileceğinin somut bir örneğidir. Gelecekte benzeri yorumlanabilir derin öğrenme tekniklerinin, özellikle bilimsel araştırmalarda ve kritik karar destek sistemlerinde daha fazla rol alması beklenmektedir.

**Kaynaklar:** Makaledeki referanslara göre derlenmiş olup, soruda belirtilen içeriğin kaynak numaraları cevap içinde ilgili cümlelerin sonunda sunulmuştur. <sup>12 15 30 33 46 52 53 63 72 78 100 110 126 131 132 134</sup>

1 2 [arxiv.org](https://arxiv.org)

<https://arxiv.org/pdf/2406.16708>

3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31  
32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60  
61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89  
90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116  
117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 [2406.16708]

CausalFormer: An Interpretable Transformer for Temporal Causal Discovery

<https://arxiv.org/html/2406.16708>